2013

Resource management algorithms for real-time wireless sensor networks with applications in cyber-physical systems

Benazir Fateh
Iowa State University

Resource management algorithms for real-time wireless sensor networks with applications in cyber-physical systems

by

Benazir Fateh

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Manimaran Govindarasu, Major Professor
Zhengdao Wang
Venkataramana Ajjarapu
Daji Qiao
Sarah Ryan

Iowa State University

Ames, Iowa

2013

To my parents Tasadduq and Khadija Fateh and husband Firoz Jaipuri for their love, patience and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

In the name of God, the most merciful and the most compassionate. I am grateful to God for giving me a life full of happiness, hope and opportunities and I ask for his blessings always. I am thankful to God for blessing me with the company of many people who have impacted my life in an amazingly positive way. In the following, I would like to acknowledge them.

First and foremost, I would like to thank Dr. Manimaran Govindarasu. I am deeply grateful to him for the immense research insight, brilliant inspiration and invaluable guidance that he has provided me during the course of my graduate studies. I greatly appreciate his constructive criticism, critical feedback and pursuit of perfection in all of my research projects. I admire his patience and kindness with me. I am thankful for the opportunities that were presented to me by Dr. Manimaran over these past years in order to make me a better student, a better researcher and a better person at large. I will continue to be inspired by Dr. Manimaran and the values of persistence effort, courage of conviction and dedication towards work that he exhibits through his work.

I express my sincere gratitude to Dr. Zhengdao Wang. In various projects, I have immensely benefitted from his knowledge and research insight. I am thankful to him for the numerous technical discussions I have had with him and the reviews that he has provided for my manuscripts.

I would like to thank Dr. Venkataramana Ajjarapu for his expert advice, collaboration and paper review for my research project. I would like to thank my committee members Dr. Daji Qiao and Dr. Sarah Ryan for being excellent teachers. I have greatly benefitted from the courses taught by them and their suggestions and insight for my research projects.

I would like to thank my senior labmate Dr. G. S. Anil Kumar for being very helpful during

the start of my graduate career. His research set an excellent example for me and frequent discussions with him have helped shape my research from time to time.

I would like to thank my labmates Dr. Adam Hahn, Dr. Naeem Al-Oudat, Haribabu Narayanan, Kayalvizhi Manimaran, Aditya Ashok, Siddharth Sridhar, Dr. Mohammad Fraiwan, Dr. Muthuprasanna Muthusrinivasan and Ki Sung Koo for interesting discussions and for making the lab environment fun. Their feedback on manuscripts and presentations has been immensely useful.

I would like to thank my friends Madhu Monga, Dr. Olga Nikolova, Swati Bansal, Mohammad Baldiwala, Sharon Sagar, Anupreet Kaur, Savitha Bangalore, Ritushree Chatterjee and Navneet Malani for being my great friends. At various times, I have drawn inspiration from each one of them. I would like to thank my group of friends in Ames who have always supported me and have made my life in Ames ever so wonderful.

Last but never the least, I would like to thank my family for making me who I am today. I am grateful to my father, Tasadduq Hussain Fateh, for having high expectations of me and teaching me the value of work early on in life; my mother, Khadija Fateh for being my role model and helping me find strength in trying times; and my husband, Dr. Firoz Ali Jaipuri for being my friend and guide for the past decade. I draw on his patience and mental strength. Finally, I would like to thank my grandmother, Shaheen, Ali Zeeshan and my whole family for their love and constant encouragement.

ABSTRACT

Wireless Sensor Networks (WSN) are playing a key role in the efficient operation of Cyber Physical Systems (CPS). They provide cost efficient solutions to current and future CPS requirements such as real-time structural awareness, faster event localization, cost reduction due to condition based maintenance rather than periodic maintenance, increased opportunities for real-time preventive or corrective control action and fine grained diagnostic analysis. However, there are several critical challenges in the real world applicability of WSN. The low power, low data rate characteristics of WSNs coupled with constraints such as application specified latency and wireless interference present challenges to their efficient integration in CPSs. The existing state of the art solutions lack methods to address these challenges that impediment the easy integration of WSN in CPS.

This dissertation develops efficient resource management algorithms enabling WSNs to perform reliable, real-time, cost efficient monitoring. This research addresses three important problems in resource management in the presence of different constraints such as latency, precedence and wireless interference constraints. Additionally, the dissertation proposes a solution to deploy WSNs based real-time monitoring of critical infrastructure such as electrical overhead transmission lines.

Firstly, design and analysis of an energy-aware scheduling algorithm encompassing both computation and communication subsystems in the presence of deadline, precedence and interference constraints is presented. The energy-delay tradeoff presented by the energy saving technologies such as Dynamic Voltage Scaling (DVS) and Dynamic modulation Scaling (DMS) is studied and methods to leverage it by way of efficient schedule construction is proposed. Performance results show that the proposed polynomial-time heuristic scheduling algorithm

offers comparable energy savings to that of the analytically derived optimal solution.

Secondly, design, analysis and evaluation of adaptive online algorithms leveraging run-time variations is presented. Specifically, two widely used medium access control schemes are considered and online algorithms are proposed for each. For one, temporal correlation in sensor measurements is exploited and three heuristics with varying complexities are proposed to perform energy minimization using DMS. For another, an adaptive algorithm is proposed addressing channel and load conditions at a node by influencing the selection of either low energy or low delay transmission option. In both cases, the simulation results show that the proposed schemes provide much better energy savings as compared to the existing algorithms.

The third component presents design and evaluation of a WSN based framework to monitor a CPS namely, electrical overhead transmission line infrastructure. The cost optimized hybrid hierarchical network architecture is composed of a combination of wired, wireless and cellular technologies. The proposed formulation is generic and addresses constraints such as bandwidth and latency; and real world scenarios such as asymmetric sensor data generation, unreliable wireless link behavior, non-uniform cellular coverage and is suitable for cost minimized incremental future deployment.

In conclusion, this dissertation addresses several challenging research questions in the area of resource management in WSNs and their applicability in future CPSs through associated algorithms and analyses. The proposed research opens up new avenues for future research such as energy management through network coding and fault diagnosis for reliable monitoring.

CHAPTER 1 Introduction

Cyber-physical systems have highly distributed and computationally intensive processing requirements along with a demand for timely and reliable wireless networking for effective monitoring and control. Wireless Sensor Networks (WSNs) are capable of meeting these requirements in a cost-efficient manner and hence have found applications in diverse areas like condition monitoring of critical infrastructure [4], [5], road/traffic monitoring [6] for surveillance, industrial automation, weather monitoring and intrusion detection to name a few.

1.1 Cyber-physical systems

Cyber-Physical Systems (CPSs) are complex integration of interdependent computational and physical processes. Identifying, understanding, and analyzing such interdependencies pose significant challenges which are greatly magnified by the geographical expanse and complexity of individual infrastructures and the nature of coupling among them [7]. CPSs represent a bold new generation of systems that integrate computing and communication capabilities with the dynamics of physical and engineered systems. A gamut of emerging applications fall in the CPSs category which take computation, communication, monitoring and control to new levels of complexity. These applications connect the physical systems embedded with low power wireless nodes (integrated with multimodal sensors) to the cyber systems which are responsible for communication and control. Emerging applications include Smart Grid [8], [9], condition monitoring of critical infrastructure like power delivery systems [4], automated traffic control, ubiquitous healthcare monitoring etc.

1.2 Role of Wireless Sensor Networks in CPSs

These new application scenarios envision smart monitoring and control through the deployment of hundreds of networked wireless sensors in real time environments. These systems with integrated sensing, computation and communication capabilities are the core of CPSs. WSNs are capable of real time monitoring over vast geographical areas. Wireless sensor based monitoring provides several advantages like real-time structural awareness, cost reduction due to condition based maintenance rather than periodic maintenance, increased operator awareness, increased safety and reliability and improved capability for forensic and diagnostic analysis etc. The data communication infrastructure involves processing a vast number of data transactions for analysis and automation. Managing the communication burden and resulting data latency is essential for efficient analysis and fast control responses and calls for distribution of intelligence throughout the infrastructure [10]. Critical infrastructures such as the electrical transmission grid are complex physical and cyber-based systems that form the lifeline of modern society, and their reliable and secure operation is of paramount importance to national security and economic vitality.

1.3 Resource management in WSN

WSNs are resource constrained with respect to energy, bandwidth, memory and processing power. While the collective power is what makes WSNs useful in real world applications, the individual stringent capabilities of each sensor pose significant challenges. In this context, the problem of energy-aware resource management in networked real-time WSNs and their integration in CPS applications is addressed.

This research refers mainly to a sensor network model consisting of one sink node or base station and a large number of sensor nodes deployed over a large geographic area. Data is transferred from sensor nodes to the sink through a multi-hop communication paradigm. Resource management algorithms for WSNs and their integration with future applications will need to take three types of factors into account: first, the resource limitations of a single sensor node with respect to battery, bandwidth and processing power; second, factors affecting

schedulability when a network of sensors is considered such as interference due to shared wireless channel and precedence constraints; and third, application dependent factors such as the latency requirements, topology specifications and any runtime opportunities presented by the physical system being monitored etc.

With respect to the first and second factors, resource management needs to happen at the individual node level and at the network level. Energy is a very critical resource and therefore, energy conservation is a key issue in the design of systems based on wireless sensor networks. Low battery power can not only reduce network lifetime but it gives rise to random and uncertain sensor measurements which unless identified jeopardize the very purpose of monitoring. A taxonomy of approaches for energy management in static WSNs is shown in Fig. 1.1.

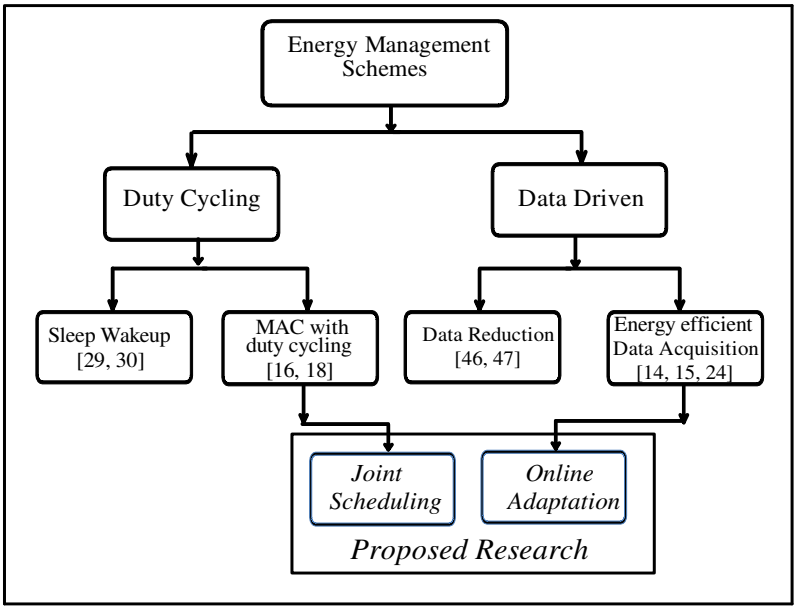

Figure 1.1: Taxonomy of existing energy management algorithms

Given that computation and communication subsystems account for the major energy consumption, both the subsystems need to be looked at for any energy minimization solution. But the solution approach of all the studies mentioned above do not take into account the computation task scheduling performed by each node along with the scheduling of communication messages. Prior work on scheduling of computation and communication subsystems

either focuses on different objective functions [11] or does not take into account computation task scheduling [12] or does not take into account the wireless interference [13].

Additionally, scheduling protocols in WSNs can leverage runtime opportunities offered by the underlying system being monitored. Specifically, such conditions may arise due to correlation in the attributes of the underlying physical system being monitored, wireless channel conditions, workload variations etc. Such runtime variations can be adaptively exploited at runtime to attain further energy minimization. Given dense deployments of WSN and shared wireless channel, typically two types of channel access schemes are used: Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access/ Collision Avoidance (CSMA/CA). The existing work in this direction focuses on centralized approaches such as adaptive sampling [14, 15] for TDMA systems and contention window adaptation methods like [16], [17] for CSMA/CA systems; or channel aware variable rate methods such as [18]. WSNs are used widely for monitoring applications where the main purpose is continuous data collection of some physical parameters such as temperature, tilt, environmental pressure etc. Since sensor data streams are measurements of continuous physical phenomenon, spatial and temporal correlations within data streams are inherent and can be effectively leveraged.

Future applications require a high-performance data communication network that supports operational requirements like real-time monitoring and control. One such application is transmission line infrastructure. WSNs have been proposed to improve the state of the art in transmission line monitoring for real-time monitoring and control [3, 4, 19–22]. While WSNs are a promising solution, they are limited by low data rate constraints which cannot keep up with the low latency, high data rate requirements. Current research either addresses this challenge either at a very high level of abstraction [4, 19] or makes unrealistic assumptions about the underlying system [3].

1.4 Research Questions Addressed in the Thesis

Given the existing research in this area of focus, several research gaps can be identified. This thesis contributes by providing solutions to those unaddressed problems. Fig. 1.2 summarizes

the thesis contributions.

Firstly, at the system level and at the network level, the need for a comprehensive scheduling scheme with the objective of energy minimization is identified. A comprehensive scheduling scheme should address joint scheduling of computation and communication subsystems and should also address the wireless interference constraints. The joint scheduling problem poses challenges, for example, the different sized time slots required by task execution and message transmission prevents the use of graph coloring approaches where each color denotes a fixed sized time slot. This research observes that the assumption of monotonically decreasing energy consumption of messages employing dynamic modulation scaling is incorrect and proposes taking the nature of the energy-delay curve into consideration before allocating slack to messages. The second chapter addresses these challenges and explains the design and evaluation of an offline algorithm that achieves joint scheduling of tasks and messages in data collection tree based WSNs to minimize energy consumption.

Secondly, it is observed that runtime variations either presented by runtime channel conditions or due to conditions of the underlying physical system being monitored, may result in unused resources thus presenting an opportunity for further optimization. Such runtime variations can be exploited using online adaptive algorithms as described in the third chapter. The design and evaluation of online algorithms that cater to two widely used Medium Access Control schemes namely, TDMA and CSMA/CA is presented.

The final component of this research addresses issues related to the applicability of WSNs in CPS application. Any future application aiming to utilize WSN presents its own set of unique specifications and challenges. Monitoring transmission lines using WSNs is a significantly different problem than monitoring other structures using WSNs owing to the specifications such as specific chain topology traversing a large geographical area. In order to address this, a hybrid hierarchical network design and evaluation is presented in the fourth chapter. The existing research in this area [3] relies heavily on symmetry. The underlying network infrastructure and the cellular infrastructure is assumed to be symmetric and available at all times. Further, it is assumed that all transmission towers are identical and transmit the same amount of data.

This creates a motivation to devise a network design solution that addresses each one of the asymmetries and forms the third component of the proposed research addressing these real world challenges.

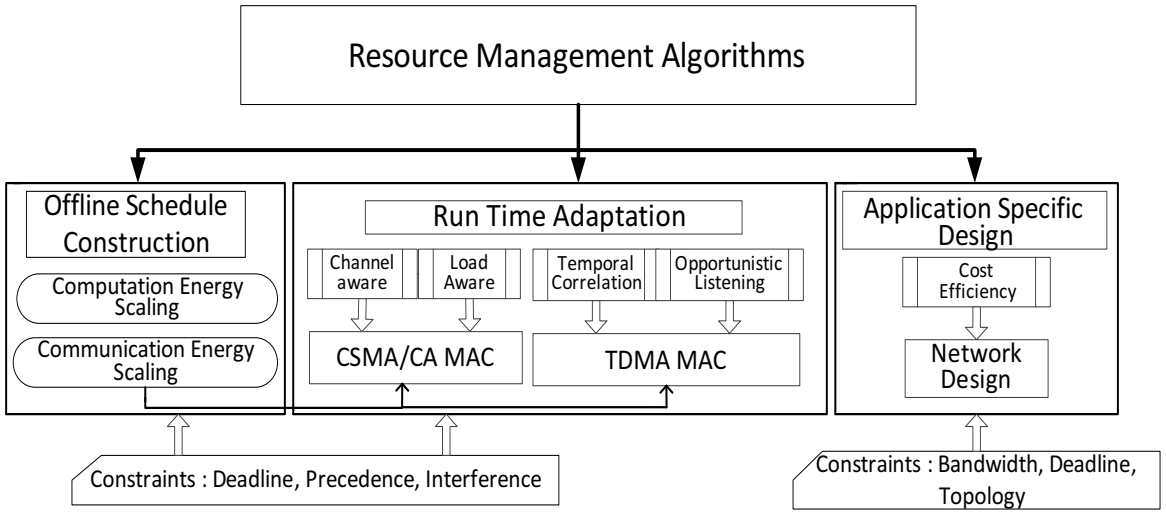

Figure 1.2: Thesis Contributions: Resource Management Algorithms

Fig. 1.2 summarizes the research contributions made through this thesis. The unifying theme of this thesis is presenting designs and algorithms that help in efficient operation of WSN thus enabling their seamless integration with the future CPS applications despite the challenges of limited resources faced by WSN. Specifically, the first two research components focus on energy management in WSN through scheduling approaches. The first part is offline energy-aware schedule construction targeted at tree based data collection applications which commonly employ TDMA as the channel contention scheme. The second research component focuses on further minimizing energy consumption at runtime by exploiting runtime variations. We propose adaptive runtime algorithms for two widely used channel contention schemes. Our schemes propose energy efficient operation of WSN which results in longer lifetimes of the sensor nodes. In the third component, we focus on a transmission line infrastructure monitoring application and address issues related to the real world applicability of WSN in such real-time applications. The algorithms proposed in part one and two can be applied for the energy management of sensors used to monitor transmission lines in part three. Energy management might not be a critical design challenge for the problem proposed in part three, where sensors

can alternatively harvest energy from ambient sources. However, other resource constraints faced by WSN such as limited wireless bandwidth, application latency and monetary cost present more dominant challenges in that research component.

1.5 Thesis Statement

The combination of limited resources and power, network topologies, time-varying wireless channel, temporal requirements, dynamic workloads and high reliability requirements presents unique challenges towards efficient operation of WSN. The goal of the proposed research is to devise efficient resource management algorithms for energy-constrained and highly dynamic wireless sensor networks in order to support easy integration with future applications. This research addresses challenges like attaining energy efficiency in the presence of wireless interference, precedence and real-time constraints. Resource management is proposed in resource constrained WSNs by way of joint computation-communication scheduling mechanisms, adaptive load and channel aware transmission scheduling schemes and slack management schemes exploiting temporal redundancy. For reliable real-time monitoring of geographically dispersed electrical overhead transmission lines, a hybrid wireless network for real-time delivery of sensor data is designed.

1.6 Thesis Organization

The rest of the dissertation is organized as follows.

- In Chapter 2, an algorithm for joint scheduling of computation tasks and communication messages in data collection tree based networks with the objective of energy minimization is presented. A Mixed Integer Linear Program (MILP) to find an optimal solution is formulated. Then, a three phase heuristic is proposed which first performs joint scheduling of tasks and messages and then reduces the energy consumption of the network by using the energy saving techniques namely Dynamic Voltage Scaling (DVS) for tasks and Dynamic Modulation Scaling (DMS) for messages. Simulation results are presented towards the end of the chapter.

- Chapter 3 proposes online scheduling in TDMA and CSMA/CA systems. For TDMA systems, a lightweight node level slack identification method exploiting temporal correlation amongst sensor measurements is proposed. Further, heuristics with varying complexities are proposed and evaluated with the objective of energy minimization while utilizing DMS. For CSMA/CA systems, a load aware and channel aware heuristic that attains joint reduction in energy and delay at the node level is proposed and evaluated.

- Chapter 4 presents hybrid network design for real-time monitoring of electrical overhead transmission lines using WSN. A formulation for designing a high performance data communication network with the objective of minimizing the installation and operational costs while satisfying the end-to-end latency and bandwidth constraints of the data flows is presented. A placement problem is formulated to find the optimal location of cellular enabled transmission towers and evaluation results of the optimization solution are presented for diverse scenarios. The proposed formulation is generic and addresses real world scenarios with asymmetric sensor data generation, unreliable wireless link behavior and non-uniform cellular coverage etc.

- Chapter 5 presents conclusions and identifies several interesting research problems in the area of resource management in WSNs.

CHAPTER 2 Joint Scheduling of Tasks and Messages for Energy Minimization in Interference-aware Real-time Sensor Networks

2.1 Summary

Emerging applications of wireless sensor networks requiring extensive in-network information processing and communication call for algorithms for joint scheduling of tasks and messages in an energy efficient manner. Dense deployments of wireless nodes and shared wireless channel pose severe interference constraints. Variation in the task and message workload prevents the traditional fixed slot size scheduling algorithms to be applied here. Several scheduling schemes in literature propose interference-aware message scheduling with the objective of energy minimization, but the problem of joint scheduling of tasks and messages for energy minimization in interference-aware manner has not been studied. In this chapter, a Mixed Integer Linear Program (MILP) is formulated for the joint scheduling of computation tasks and communication messages in data collection tree based networks. Then, a three phase heuristic is proposed which first performs joint scheduling of tasks and messages and then reduces the energy consumption of the network by using the energy saving techniques namely Dynamic Voltage Scaling (DVS) for tasks and Dynamic Modulation Scaling (DMS) for messages.

2.2 Background

The massive scale at which the sensor networks could be deployed demand smart usage of the available energy to reduce the cost of monitoring. The future applications have highly

distributed and computationally intensive processing requirements along with a demand for timely and reliable wireless networking for effective monitoring and control. These applications leverage onboard computational capability on the wireless sensors to allow data processing to occur within the network. The individual computed results are then wirelessly transmitted. Coexistence of extensive computation and communication demand integrated scheduling schemes.

Computation and communication subsystems dissipate bulk of energy in a sensor node. Recent research [23] and [24] study the problem of system-wide energy management by distribution of unused time slots (slack) amongst the computation and communication subsystems using energy saving techniques like Dynamic Voltage Scaling (DVS) and Dynamic Modulation Scaling (DMS). The DVS technique saves computation energy by simultaneously reducing CPU supply voltage and frequency. The DMS technique saves communication energy by reducing the radio modulation level [25]. In [23] and [24], DVS and DMS are considered to be analogous which is not always correct. In DVS, energy is always a monotonically decreasing function of delay but this is not true in the case of DMS.

The energy consumed by the radio of a wireless device is made up of two components: the transmission energy and the circuit consumption energy. The transmission energy is a function of several variables like the transmitter-receiver distance, transmission time, channel gain and atmospheric noise. The circuit consumption energy on the other hand is a linear function of the time the transmitter and receiver circuit need to be on. For the transmission energy calculation, consider M-ary Quadrature Amplitude Modulation (MQAM) as the modulation scheme. In MQAM, there are $\mathcal{M}$ constellation symbols, with b bits per symbol such that $\mathcal{M}$ is equal to 2^b. Changing the modulation levels implies varying the number of bits in each symbol. Decreasing(increasing) the modulation level implies reducing(increasing) the number of bits in each symbol. This requires transmitting more symbols. Given a constant symbol rate, more symbols means more transmission time.

As noted in [26], in traditional wireless networks, the transmission distance is large($\geq 100m$), hence the transmission energy is dominant in the total energy consumption. There-

fore, in such wireless networks the main objective is to minimize the transmission energy. However, in sensor networks the nodes are densely distributed, and the average distance between nodes is usually below $10m$. As the modulation level decreases and the transmission time increases, the transmitter and receiver circuit need to be on for a longer duration. In such scenario, the circuit energy consumption along the signal path becomes comparable to or even dominates the transmission energy in the total energy consumption. Hence, at lower transmitter-receiver distance, continuously reducing the radio modulation level may lead to increased energy consumption. The study in [26] considers lower bandwidth (= 10KHz) case. We conducted simulations for higher bandwidth cases to get a more realistic scenario for presently used WSNs which use the IEEE 802.15.4 standard and have bandwidth of 3MHz. Our observations are shown in Fig. 2.1 and Fig. 2.2.

In Fig. 2.1 and Fig. 2.2, the vertical axis is the energy consumption per information bit (log_{10} $\frac{E_a}{0.001}$ dBmJ) and the horizontal axis is the normalized transmission time, ($\frac{T_{on}}{T_{max}}$). The transmission time T_{on} is the time the transmitter-receiver circuit needs to be *on*. It is a function of modulation level, b. This transmission time is normalized with respect to the maximum transmission time, T_{max} (corresponding to the lowest modulation level, b (=2)). Thus the horizontal axis represents the time the transmitter-receiver circuit need to be *on* at different modulation levels. This time increases as modulation level is decreased from 10 to 2. The points on the graph correspond to energy consumption per bit for modulation level, b which can take values from the set {10, 8, 6, 4, 2}.

Fig. 2.1 shows energy-delay curve for distances ranging from 100m to 5m. Fig. 2.2 shows energy-delay curve at smaller distances ranging from 20m to 2m. It can be seen that at higher transmitter receiver distances (100m - 50m), the total energy consumption is a monotonically decreasing function of normalized transmission time. But the same is not true for lower distances (lesser than 25m). The results show that at transmitter-receiver distance, $d \leq 20m$, modulation scaling initially yields energy savings but leads to higher energy consumption as the modulation level is decreased further. At $d \leq 5$, operating at the highest modulation level is the most energy efficient. We use these observations and perform slack distribution only

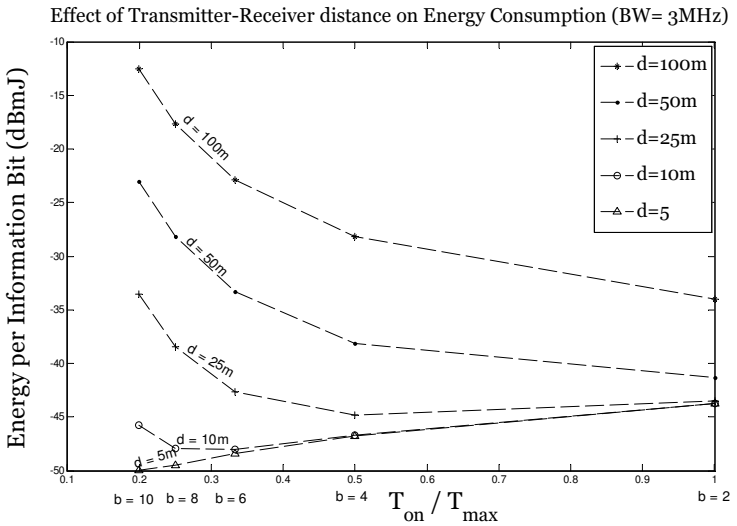

Figure 2.1: Effect of large transmitter-receiver distance

when there is a reduction in the total energy consumption.

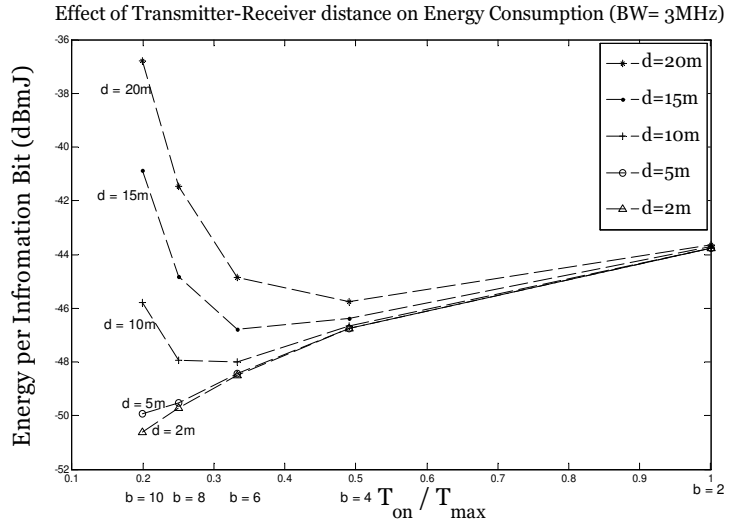

Figure 2.2: Effect of small transmitter-receiver distance

In order to provide timeliness guarantees, Time Division Multiple Access(TDMA) scheduling is best suited for data collection tree-based topologies. TDMA scheduling assigns time slots to nodes such that the data generated at each node reaches the root of the tree by the end of the frame. Two nodes can transmit in the same time slot using the same channel if their transmissions do not interfere with each other. Spatial reuse is advantageous as the schedule length can be minimized and the resulting slack can be used by DVS and DMS to reduce the

energy consumption of the network. In this chapter, we study this problem of joint scheduling and slack distribution with the objective of energy minimization.

The chapter is organized as follows: in section 2.3 we cite related research, section 2.4 presents the system model. Section 2.5 presents the MILP formulation for obtaining the optimal solution. Section 2.6 explains our proposed algorithm. Performance results are presented in Section 2.7 and section 2.8 concludes the chapter.

2.3 Related Work

Existing literature in the realm of TDMA based interference-aware link scheduling can be classified with respect to their objective functions. In some works like [27] and [28], the objective is to increase throughput and minimize delay without any consideration for the energy consumption, while studies in [29], [30] aim at minimizing the energy consumption by minimizing the energy wasted during sleep-wakeup transitions. With stronger processors, more memory on board and availability of higher bandwidth, sensor nodes are more equipped and are widely deployed to perform extensive computation tasks. Thus task scheduling should be given explicit consideration. But the solution approach of all the studies mentioned above do not take into account the computation task scheduling performed by each node.

Prior work on joint scheduling of computation and communication subsystems is presented in [11], [31], [13] and [12]. In [11], the objective is maximizing the number of clients that can be scheduled with limited processor capacity and link capacity. In [31] the interference issues are not taken into account. In [13], an energy aware computation-communication scheduling is studied where DVS is used for computation energy reduction. However, they consider a very simplistic model of one-hop network, ruling out parallel transmission and interference.

In [12], authors provide a real-time task mapping and scheduling solution in multi-hop homogeneous WSNs. However, they only leverage the energy-delay tradeoff for computation tasks using DVS, as opposed to our scheme which uses a combination of DVS and DMS. Their approach to reduce energy consumption by slack reclamation using DVS equally scales down the CPU speed of all the sensors. This approach will yield poorer results when applied to a

system where both DVS and DMS is available. This is because any two processors will yield the same energy savings when scaled down equally that is from a frequency level f_1 to f_2. But two messages might differ in their energy savings when scaled down from same modulation levels m_1 to m_2 owing to different distances that these messages need to be transmitted over. So a combination of DVS and DMS needs an iterative scheme to yield better energy savings.

System-wide energy reduction using the combination of DVS and DMS is also studied in [23], [24] and [32]. But in all these works, the interference issues are not considered. Also the assumption of monotonically decreasing energy consumption of messages employing DMS is not correct. In [32], authors investigate the problem of task allocation to a single-hop cluster of homogeneous sensor nodes with multiple wireless channels with the objective to minimize the balanced energy consumption of the network. The multiple wireless channel model can serve to solve the interference issues only in small sized networks ($\approx$10 nodes) as considered in [32]. In large scale WSNs, interference issues need explicit consideration. On a different note, environmental resource harvesting schemes [33] can reduce the problem of limited energy budget but they have low unstable efficiency due to uncontrollable environmental conditions rendering them unsuitable for real-time applications.

To the best of our knowledge, this is the first work that addresses the joint scheduling of communication and computation to minimize energy consumption in an interference-aware network. The joint scheduling problem poses several challenges. The time taken by the different task execution and message transmission are different, preventing the use of graph coloring approaches where each color denotes a fixed size slot. Time intensive computation tasks must be executed at the node itself. They cannot be offloaded to a distant server owing to the large energy consumption of the wireless transmitted messages. Additionally, the precedence and interference conflicts present among the scheduled entities prevent the optimal allocation of resources in order to reduce energy consumption.

The main contributions of this chapter are:

1. We formulate the Interference-aware Energy-aware Joint Scheduling (IEJS) problem. Constructing an energy-aware schedule which minimizes energy consumption, considering the con-

vex nature of the energy-time tradeoff curve and the various conflicts present is an NP-hard problem. It can be shown that the Multiple Choice Knapsack Problem (MCKP) can be polynomially reduced to the energy-aware scheduling problem.

2. We take the nature of the energy-delay curve into consideration before allocating slack to messages. As we show in Fig. 2.1 and Fig. 2.2, incremental slack allocation is not always energy efficient for messages. Further, when tasks and messages are jointly considered, there are two types of entities (tasks and messages) contending for the available slack. These two entities differ in the way they interfere with objects of their same type. Two message transmission (not bounded by precedence constraints) can interfere with each other due to spatial proximity of the involved transmitters and receivers. But that is not the case with tasks. Any two tasks can be simultaneously scheduled as long as there is no precedence relation between the two.

3. We present a mixed integer linear programming (MILP) formulation of our problem. The optimal solution of the problem can be obtained by using a commercial software package such as [34], though it is computationally very expensive. We develop a polynomial time algorithm for joint scheduling of tasks and messages in a variable TDMA setting. Our method takes into consideration the interference and precedence constraints and the varying nature of tasks and messages.

Problem Statement: The problem of interference-aware energy-aware joint scheduling of tasks and messages (IEJS) that we study in this chapter can be stated as:

IEJS: *Given a set of tasks and messages allocated to nodes in a data aggregation tree, obtain a feasible schedule for the tasks and messages that minimizes the energy consumption of the network while preserving the precedence, deadline and interference constraints.*

The $IEJS$ problem can be shown to be NP-hard by polynomial reduction of the Multiple Choice Knapsack Problem (MCKP) to the energy-aware scheduling problem.

2.3.1 Problem Complexity

We prove the complexity of the problem by reducing the Multiple Choice Knapsack problem (MCKP) to an instance of the Interference-aware Energy-aware Message Scheduling (IEMS)

problem. The MCKP problem is stated as: Given k classes N_1,...N_k of items to pack in a knapsack of weight capacity C, where each item $O_{ij} \in N_i$ has a value v_{ij} and weight w_{ij}, the problem is to choose exactly one item from each class such that the value sum is maximized without having the weight sum to exceed C. Fig. 2.3 shows an illustration of the MCKP and its reduction to the energy-aware scheduling problem. Fig. 2.3 shows three classes of items with each item having a fixed weight and value. The MCKP problem is to choose exactly one item from each class such that the value sum is maximized without the weight sum to exceed the weight capacity, C. The MCKP can be mathematically expressed as follows:

Maximize:

$$\sum_{i=1}^{k} \sum_{j \in N_i} v_{ij} x_{ij} \tag{2.1}$$

Subject to:

$$\sum_{i=1}^{k} \sum_{j \in N_i} w_{ij} x_{ij} \leq C \tag{2.2}$$

$$\sum_{j \in N_i} x_{ij} = 1 \quad \forall 1 \leq i \leq k \tag{2.3}$$

$$x_{ij} \in \{0, 1\} \quad \forall 1 \leq i \leq k \ \ and \ \ j \in N_i \tag{2.4}$$

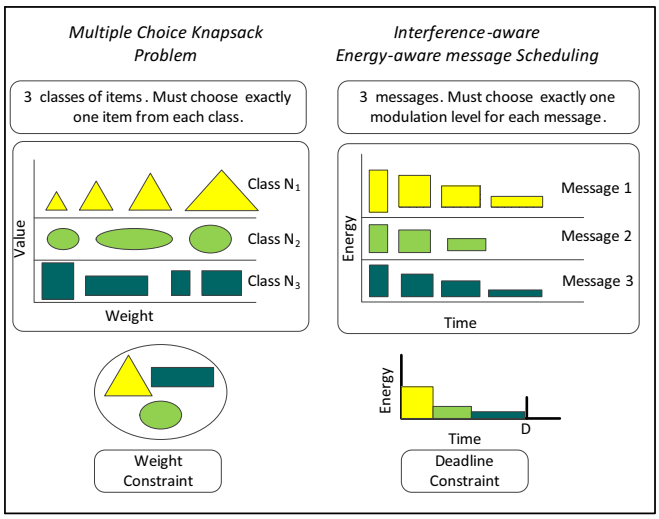

Figure 2.3: Reduction of MCKP to IEMS

Theorem 1: The problem of scheduling a set of messages with the goal of minimizing the total energy consumption while satisfying the interference, precedence and deadline constraints is NP-hard.

Proof. For the convenience of reduction, we rewrite the maximization objective of the MCKP as the following minimization objective:
Minimize: $\sum_{i=1}^{k} \sum_{j \in N_i} -v_{ij} x_{ij}$
Now, the MCKP has a one to one correspondence with the $IEMS$ problem and can be reduced to an instance where maximum interference is considered. In the case of maximum interference, a message transmission interferes with every other message reception in the network. Hence, message scheduling has to be completely sequential barring any parallelism in message transmission. Note that this instance of sequential message transmission is considered for the sake of reduction to the MCKP and this simplest case of sequential message transmission is also found to be hard. Thus, in cases where there is less interference and there is an opportunity for spatial reuse, the scheduling is going to be even harder. For the purpose of reduction, create a message M_i for each of the classes N_i. Each message can be transmitted at one of the modulation levels MO_{ij}. For each object O_{ij} in N_i create a modulation level MO_{ij} for M_i. Now the weight of the object, w_{ij}, can be mapped to the time, T_{ij} taken by the message's modulation level and the value of the object, v_{ij}, can be mapped to the energy consumption, E_{ij}, of the message's modulation level. The weight constraint of the knapsack can be mapped to the deadline constraint in the scheduling problem as all the messages are sequentially transmitted. If M is the total number of messages and λ_m is the number of modulation levels available to each message, then this reduction takes $O(M\lambda_m)$ amount of time which is a polynomial of the problem size. The $IEMS$ problem can be formally stated as:

Minimize:

$$\sum_{i=1}^{k} \sum_{j \in M_i} E_{ij} y_{ij} \tag{2.5}$$

Subject to:

$$\sum_{i=1}^{k} \sum_{j \in M_i} T_{ij} y_{ij} \leq D, \tag{2.6}$$

$$\sum_{j \in M_i} y_{ij} = 1, \quad \forall 1 \leq i \leq k \tag{2.7}$$

$$y_{ij} \in \{0,1\} \quad \forall 1 \leq i \leq k \ \ and \ \ j \in M_i \tag{2.8}$$

Refer to Fig. 2.3 for an illustration of the reduction mechanism. Each class can be mapped to a message and items in each class can be mapped to the message at different modulation levels having different energy consumption and delay. In order to schedule these messages, exactly one modulation level of each message must be picked. This implies that if the interference-aware energy-aware message scheduling problem can be solved in polynomial time, then the MCKP can also be solved in polynomial time. However, it is known that the MCKP problem is NP-hard. Therefore, the scheduling problem is also NP-hard. □

Now in the existing $IEMS$ problem, if we add another constraint which is energy-aware task scheduling, then the problem becomes $IEJS$ and it remains NP-hard. Thus, the joint scheduling of tasks and messages with interference, precedence and deadline constraints is NP-hard.

2.4 System Model

In data monitoring applications, each node in the network is required to perform some periodic tasks like sensing, computation and communication. A leaf node processes the data sampled by its sensor suite. All the children nodes transmit their data to the parent node. Once the parent node receives data from all its children, it performs data aggregation and transmits this aggregated message to its parent node. Thus given a fixed tree topology where the nodes are static, the problem of task mapping has been taken care of and the problem of scheduling remains. At a node v_i, the computation task is denoted by t_i and the communication message is denoted by m_i. In the rest of the chapter we use the term *entity* to refer to t_i and m_i unless explicitly stated. Nodes v_j = {1,2...,k} communicating to the same parent node, H are henceforth said to be in a cluster with the parent node referred to as cluster head. Any two

nodes in a cluster cannot transmit simultaneously since the receiver (cluster head) can only correctly receive signal from one transmitter at a time.

2.4.1 Interference Model

In wireless networks, the message transmitted by a node may be received by all the nodes within its transmission range. Usually two types of ranges are associated with each transmitter: a transmission range (r) and an interference range (R), with $R \geq r$. In order to receive a message and successfully decode it, one of the necessary condition is that the receiver must be in the transmission range of the transmitter. When a node v_i is outside the transmission range but inside the interference range of a transmitter v_j, node v_i cannot receive the signal from v_j successfully, but the signal can corrupt any other transmission intended for v_i which v_i could have otherwise successfully decoded. There are several models proposed for interference modeling. We use the protocol model proposed in [35] for interference modeling in wireless networks. In the protocol model, interference range is a function of the transmitter-receiver distance. According to the protocol model [35], if a node v_i transmits to a node v_j, then this transmission is successfully received by node v_j if

$$| v_j - v_k | \geq (1 + \delta) | v_j - v_i | \tag{2.9}$$

for any node v_k simultaneously transmitting to any node v_m at the same time using the same channel. The interference range is a function of the parameter δ and the transmitter receiver distance. It is defined with respect to a link. In data aggregation tree, a node can have more than one child node and hence has different interference ranges with respect to each of the child node as shown in Fig. 2.4. To correctly model the different interference ranges, we model each link as a node in the mixed graph as explained later. And an edge exists between two nodes in the mixed graph if the two links in the original graph cannot be scheduled simultaneously. Protocol model implicitly assumes that each node v_i will adopt the power control mechanism while transmitting.

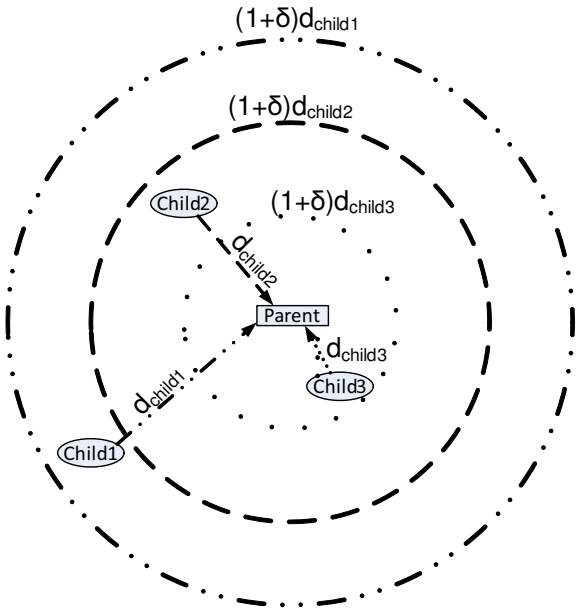

Figure 2.4: Different interference ranges at parent node

2.4.2 Computation-Communication Model

Each node supports DVS with λ_t discrete frequency levels and DMS with λ_m discrete modulation levels. We consider a DVS-enabled computation subsystem where F denotes the set of available discrete frequencies and P denotes the set of associated power levels at each of these frequencies. The ith element of these sets is denoted as F_i and P_i. Each sensor node performs a task of C computation cycles. The time taken by a task of C computation cycles is calculated as $\frac{C}{F_i}$. The associated energy at the ith performance level can then be calculated as $P_i \cdot \frac{C}{F_i}$.

For the communication subsystem, we consider M-ary Quadrature Amplitude Modulation (MQAM) as the modulation scheme. An Additive White Gaussian Noise (AWGN) channel model is used to model the wireless channel. In MQAM, there are $\mathcal{M}$ constellation symbols, with b bits per symbol such that $\mathcal{M} = 2^b$. Changing the modulation levels implies varying the number of bits in each symbol. The time taken to transmit a packet of L bits is given as $\frac{L}{Wb}$, W being the channel bandwidth. The L bits in the packet constitute the message to be transmitted along with coding overheads and modulation notification bits. The total energy expended by the communication subsystem can be mainly divided into two components:

the signal transmission energy, E_{tx} and the circuit consumption energy, E_{ckt}. The circuit consumption energy at the transmitter and receiver can further be categorized as the sum of energy consumption of various circuit blocks like power amplifiers, frequency synthesizers, Analog to Digital Converter, Digital to Analog converter, mixers and low noise amplifiers etc. Specifically, it can be divided into the energy consumed by the power amplifiers, E_{amp}; energy spent in the mode transitions (active to sleep and sleep to active), E_{tr} and the energy consumption by the rest of the blocks can be combined as transmitter and receiver circuit consumption energy E_c. While E_{amp} and E_c are functions of the transmission time, E_{tr} is the constant energy consumption in the transient mode. This is mainly dominated by the frequency synthesizers and is independent of the packet size or the transmission time. Energy consumption by the power amplifier is given by, $E_{amp} = \alpha\ E_{tx}$ where α is dependent on the modulation scheme and the associated constellation size [26]. Thus the energy expended in transmitting a packet of L bits can be given as:

$$E_L = E_{tx} + E_{amp} + E_c + E_{tr} \tag{2.10}$$

The circuit consumption power, P_c can be computed as a constant for a given sensor node. The circuit consumption energy, $E_c = P_c T_{on}$ then depends only on the time the circuit needs to be on, T_{on}. The energy consumption per packet can be written as a function of the transmission time as

$$E_L = ((1+\alpha)P_{tx} + P_c)T_{on} + E_{tr} \tag{2.11}$$

where P_c is the circuit power, P_{tx} is the transmission power and T_{on} is the transmission time. Given the bit error probability, P_b, the necessary received power for successful reception can be approximated as [36],

$$P_r = \frac{4}{3} N_f \sigma^2 (\mathcal{M} - 1) \cdot W \cdot ln(\frac{4(1 - \frac{1}{\sqrt{2^b}})}{b.P_b}) \tag{2.12}$$

Refer to Table 1 for the meaning of relevant symbols. Considering a κth power path loss model at distance d, the transmission power can be calculated as $P_{tx} = P_r A_d$, where $A_d = A_g d^{\kappa} L M_l$ [26]. We assume $\kappa = 3.5$, antenna gain, $A_g = 3.3$dB and $LM_l = 40$dB represents the link margin compensating the hardware process variations and other additive background

noise or interference. Given a network of V nodes, with each node v_i having a computation task processed at frequency $F_{i,t}$ and the communication message transmitted at modulation $b_{i,m}$, the total energy consumption of a network can be calculated as follows:

$$\sum_{v_i \in V} (((1+\alpha)P_{tx_{i,m}} + P_c)\frac{L}{W \cdot b_{i,m}} + E_{tr}) + \sum_{v_i \in V} P_{i,t}\frac{C}{F_{i,t}} \tag{2.13}$$

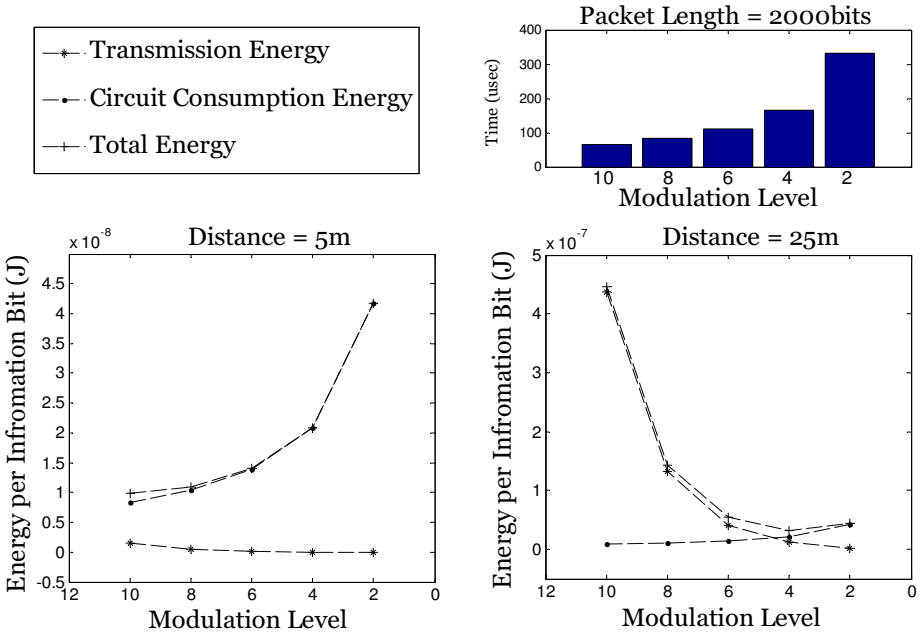

Figure 2.5: Effect of transmitter receiver distance on energy components

Fig. 2.5 shows how distance and modulation level effect the relationship between energy components (transmission energy and circuit consumption energy). At distance d = 5m, modulation scaling leads to increased energy consumption due to (predominantly) increasing circuit consumption energy. But at d = 25m, the effect is reversed. Now modulation scaling results in decreased energy consumption due to (predominantly) decreasing transmission energy. However, note that distance is the more dominant factor (since it contributes exponentially to the transmission energy calculation). Hence the lowest modulation at lesser distances will consume less energy than the highest modulation at a higher distance.

Table 2.1: Relevant Symbols

E_{tx}	Signal Transmission Energy	A_g	Antenna gain
E_{amp}	Power amplifier energy	b	Modulation level
E_c	Circuit Energy	$\mathcal{M}$	Modulation symbol
E_{tr}	Transient mode energy	W	Channel bandwidth
P_{tx}	Transmitter Power	C	Computation cycles
P_r	Receiver power	P_b	Bit error probability
P_c	Circuit power	T_{on}	Signal Transmission time
σ^2	Power spectral density	N_f	Receiver Noise figure

2.5 MILP Formulation

The interference-aware energy-aware joint scheduling problem (IEJS) can be formulated as a Mixed Integer Linear Program (MILP) to find the optimal energy consumption of the network. The formulation presented below uses the following variables: $S_{i,t}$, $S_{i,m}$, $M_{i,j}$, $T_{i,j}$ and $I_{i,k}$. The continuous variables $S_{i,t}$ and $S_{i,m}$ denote the start times of task t_i and message m_i respectively. The integer (binary) variables $M_{i,j}$ and $T_{i,j}$ denote the jth performance level chosen for the message m_i and task t_i respectively. Specifically, the variable $M_{i,j}$ is set to one if the jth modulation level is assigned to message m_i, otherwise, it is set to zero. Similarly, the variable $T_{i,j}$ is set to one if jth frequency level is assigned to task t_i, otherwise, it is set to zero. The following MILP formulation outputs a schedule that optimally minimizes the total energy consumption given a conflict graph while meeting the interference, precedence and end-to-end deadline constraints. The objective function sums up the energy consumptions of all the activities that a node performs including message transmission, message reception and computation. Here, λ_m specifies the number of available modulation levels and λ_t is the number of available frequencies. The MILP can be formulated as:

Minimize

$$\sum_{v_i \in V} \sum_{m=1}^{\lambda_m} ((1+\alpha)P_{tx_m} + P_c)\frac{L}{W \cdot b_m} + E_{tr}) \cdot M_{i,m} + \sum_{v_i \in V} \sum_{t=1}^{\lambda_t} P_t \frac{C}{F_t} \cdot T_{i,t} \tag{2.14}$$

Subject to:

$$S_{j,t} \geq 0, \tag{2.15}$$

Precedence Constraints:

$$S_{j,m} + \sum_{m=1}^{\lambda_m} \frac{L_j}{Wb_m} M_{j,m} - S_{i,t} \leq 0, \quad \forall v_j \in V_c(i), \tag{2.16}$$

$$S_{j,t} + \sum_{t=1}^{\lambda_t} \frac{C_j}{F_t} T_{j,t} - S_{j,m} \leq 0, \tag{2.17}$$

Deadline Constraints:

$$S_{R,m} + \sum_{m=1}^{\lambda_m} \frac{L_j}{Wb_m} M_{R,m} \leq D, \tag{2.18}$$

Interference Constraints:

$$S_{i,m} + \sum_{m=1}^{\lambda_m} \frac{L_i}{Wb_m} M_{i,m} \leq S_{k,m} + (1 - I_{i,k})Q, \tag{2.19}$$

$$S_{k,m} + \sum_{m=1}^{\lambda_m} \frac{L_k}{Wb_m} M_{k,m} \leq S_{i,m} + I_{i,k}Q, \tag{2.20}$$

Other Constraints:

$$\sum_{m=1}^{\lambda_m} M_{j,m} = 1, \tag{2.21}$$

$$\sum_{t=1}^{\lambda_t} T_{j,t} = 1, \tag{2.22}$$

$$M_{j,m}, T_{j,t} \in \{0, 1\} \quad \forall j, t, m. \tag{2.23}$$

$$I_{j,k} \in \{0, 1\} \quad \forall j, k \in V. \tag{2.24}$$

The constraint in Eq. 2.15 specifies that the start times of tasks at the nodes should be greater than or equal to the start of the schedule at time zero. Constraints in Eqs. 2.16 and 2.17 ensure the precedence relations at a node and also between a node and its children. The constraint in 2.16 ensures that the task at each node v_i in the network starts after it receives

all the messages from its children. Set of children of v_i is denoted by $V_c(i)$. Similarly, the constraint in Eq. 2.17 ensures that each message starts transmission only after the local task has finished its execution. Eq. 2.18 constraints the completion of the last message at the root node, R before the end-to-end deadline, D. It is assumed that the root node transmits a final aggregated message to a base station or a control center where further analysis is performed. Notice that, if the root node *is* the base station, then this constraint will change to state that the start time of the computation task at root + computation time $\leq$ Deadline. The constraints in Eq. 2.19 and Eq. 2.20 enforce the interference constraints. For any node v_i which has node v_k in its interference set, nodes v_i and v_k cannot be scheduled simultaneously. Either node v_k starts after v_i finishes or v_i starts after v_k finishes. The binary variable $I_{i,k}$ models these situations. When $I_{i,k} = 1$, it means that i will be scheduled before k, and when $I_{i,k} = 0$, it means k will be scheduled before i. Q is a very large number with a magnitude of order 3-4 times larger than the numbers used in the schedule. This is the $Big - M$ technique used in linear programming to model such constraints. Constraints in Eq. 2.21 and Eq. 2.22 ensure that exactly one performance level is chosen for the entities.

The above formulated MILP can be used to find the optimal energy consumption of the network. However, as the network size increases, the MILP becomes excessively computationally expensive as shown in our performance evaluation section.

2.6 Joint Task-Message Scheduling Algorithm

We now discuss our proposed heuristic for the IEJS problem. Our scheduling objective is energy minimization of the network while respecting the precedence, deadline and interference constraints. Given the intractability of the problem, we adopt a modular approach. By doing so, we decompose the problem and present the opportunity to plug-in existing solutions for each subproblem. We decompose the Interference-aware Energy-aware Joint Scheduling Heuristic ($IEJSH$) into the following three subproblems:

1. **Conversion of a given network topology to a mixed graph:** In the first step, a mixed graph representation is sought to explicitly model the tasks and messages as

nodes and capture the constraints amongst them. The mixed graph formulation uses the conflict graph approach [37], to model the interfering links in network graph to nodes in mixed graph.

2. **Initial schedule construction:** In the second step, we construct an initial schedule which follows the precedence, deadline and interference constraints. The tasks and messages are set to operate at the highest performance level which consumes the least amount of time. This way if a feasible solution exists, it is guaranteed to be found by an optimal solution when entities are at the highest performance levels. Once a feasible schedule is constructed, the amount of slack can be calculated.

3. **Slack reclamation:** Once an initial schedule is constructed, and slack availability is known, this slack can be allocated to tasks and messages such that the energy consumption of the network is minimized. Step 2 results in a slotted initial schedule such that entities are aligned in parallel. This method while suboptimal is so designed because it makes slack allocation in step 3 computationally easier to perform as discussed later. The performance levels of the tasks and messages can be scaled such that the energy consumption can be reduced.

2.6.1 Conversion of network graph to a mixed graph

The network graph is denoted as $G(V, E)$. V is the set of nodes and E is the set of edges such that $e(i, j)$ denotes that node v_i is a child of node v_j and hence transmits periodic messages to node v_j. We transform the network graph into a mixed graph to differentiate between the tasks and messages and to represent the different precedence and interference constraints. A mixed graph has both directed and undirected edges. Directed edges model the precedence constraints and undirected edges model the interference constraints. Mixed graph is denoted by $G_M = (V_M, E_M, U_M)$. The mixed graph has a node for each *entity*. The vertex set $\mid V_M \mid = 2 * \mid V \mid$ because each node in G performs two tasks, a computation task and a communication task. In a data aggregation tree, a node can have more than one child node and hence has different interference ranges with respect to each of the child node. To correctly model the

different interference ranges, we model each edge in G as a node in G_M. An undirected edge exists between two nodes in G_M if the two links in G cannot be scheduled simultaneously. The precedence and interference conflicts are modeled as follows:

- Precedence Constraints: At a node v_j, the computation task t_j must precede the communication message m_j. In the mixed graph, this is modeled as a directed edge, from vertex t_j to vertex m_j , $e(t_j, m_j)$. Also, to model the precedence constraints between children and parent, the communication message m_j of node v_j must precede its parent node v_i's computation task t_i. In the mixed graph, this is modeled as a directed edge, from vertex m_j to vertex t_i , $e(m_j, t_i)$.

- Interference Constraints: If any two directed edges $e(i, j)$ and $e(k, m)$ in the graph G interfere, then these two links cannot be scheduled simultaneously. This constraint is denoted by an undirected edge $u(m_i, m_k)$ in G_M. Similarly two nodes v_i and v_j in G, transmitting to the same destination cannot be scheduled simultaneously. This is modeled in G_M by an undirected edge between vertex m_i and m_j denoted as $u(m_i, m_j)$.

Given that the network is configured in the form of a data collection tree, the number of directed incident edges on any node is the number of children of that node. The number of undirected edges incident on a node is called its interference *degree*. Fig. 2.6 shows an illustrative example for the conversion of network graph to a mixed graph.

2.6.2 Initial Schedule Construction

A feasible schedule will assign each node of G_M a start time St_i and a finish time Ft_i while ensuring all the constraints. We have two types of nodes in the mixed graph: computation node and communication node. We define a tuple for each computation node v_i as $\{P, Pset\}$, where P denotes the priority and $Pset = \{v_k : e(v_k, v_i) \in E_M\}$ is the set of predecessors of node v_i. For each communication (message) node, v_i, the tuple takes the form, $\{P, D, Pset, Iset\}$, where D denotes the degree,and $Iset = \{v_k : | parent(v_i) - v_k | \leq (1+ \delta)| v_i - parent(v_i) |\}$ is the set of nodes interfering with v_i.

Input: Network Graph $G = (V,E)$
Output: Energy-unaware Feasible Schedule Sch_u
1. Convert G into mixed graph G_M.
2. For each computation node in G_M, make a tuple (P, $Pset$);
   For each communication node in G_M, make a tuple (P, D, $Pset$, $Iset$).
3. Sort nodes in G_M in a non-increasing order of priority, P.
4. Nodes with equal priority are sorted in non-increasing order of degree, D.
   Q = Sorted list of all nodes in G_M.
5. **while** $Q \neq NULL$ **do**
   Pick the first node v_i from Q.
   Assign the smallest integer number ϕ_i (≥ 0) to v_i such that $\phi_i > \phi_j \; \forall \; j \in Pset(i)$, and
   if *v_i is a message* **then**
   $\phi_i \neq \phi_j \; \forall \; j \in Iset(i)$
   end
   Q = {Q} - {v_i}.
end
6. C = List of nodes v_i, sorted in increasing order of their assigned number, ϕ, $SlotFinish_{\phi prev} = 0$
while $C \neq NULL$ **do**
   CC = Set of all nodes v_i having the same number, ϕ_i.
   For all nodes in CC {
   $St_i = SlotFinish_{\phi prev}$
   $Ft_i = St_i + Sl_t$, if v_i is a task
   $Ft_i = St_i + Sl_m$, if v_i is a message
   }
   if *all these nodes v_i are tasks* **then**
   $SlotFinish_{\phi i} = SlotFinish_{\phi prev} + Sl_t$
   end
   if *all these nodes v_i are messages* **then**
   $SlotFinish_{\phi i} = SlotFinish_{\phi prev} + Sl_m$
   end
   else
   $SlotFinish_{\phi i} = SlotFinish_{\phi prev} + \max\{Sl_m, Sl_t\}$
   end
   $SlotFinish_{\phi prev} = SlotFinish_{\phi i}$
   C = {C} - {CC}
end

Algorithm 1: Initial Schedule Construction

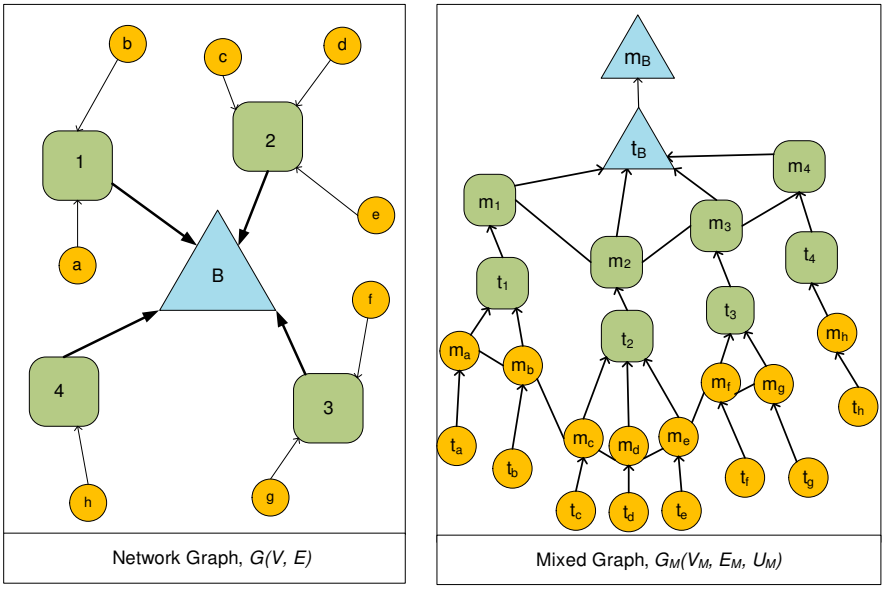

Figure 2.6: Conversion of Network Graph to Mixed Graph

The start time of a node must be greater than or equal to the finish times of all its predecessor nodes. In order to enforce the precedence constraints, we use a priority mechanism. For each node in G_M, there exists a single multi-hop path to the root. This path can be found by considering only the directed edges. *Priority* of a node is the number of hops in this path. Thus a child will always have higher priority than its parent. Nodes are now arranged in a non-increasing order of priority and selected in this order for scheduling. Due to this, a child node will be scheduled first. When the parent node is selected for scheduling, its start time must be greater than or equal to the finish time of all its predecessors (which would already be scheduled given their higher priority) thus ensuring the precedence order.

Unlike computation nodes, scheduling for message nodes requires additional considerations for interference constraints. The schedules of interfering nodes must be excluded in time. In the mixed graph, we model interference constraints among communication nodes by drawing an undirected edge between them. The number of undirected edges incident upon a message node is defined as its *degree*, D. Now, nodes with equal priority are sorted in non-increasing order of their degree and picked in this order for scheduling. We utilize the approach of ordering the message nodes in decreasing order of degree [28], so that nodes which interfere with maximum number of other nodes are scheduled first. Hence nodes scheduled later have to check for fewer

nodes before being scheduled, thus reducing the scheduling overhead. Now a node is picked from this sorted set and assigned the smallest slot not yet assigned to a conflicting entity. Algorithm 1 presents the pseudo code for the initial schedule construction.

Unlike traditional TDMA scheduling where all the slots are of same length, joint scheduling of tasks and messages requires variable length slots. At the start, all tasks and messages are assumed to be at the highest level. We define two types of time slots, Sl_t and Sl_m for tasks and messages respectively. Notice that since the data packet size for the messages and the computation cycles for tasks is constant across the whole network, given a performance level every node will require the same amount of time for task computation and message transmission respectively.

As shown in Algorithm 1, given a sorted set of entities, we first decide which entities can be scheduled in parallel, by assigning each entity a number. Later based on the composition of the set of entities having the same number, the slot length can be decided. For example, if all task nodes are aligned in parallel then the slot length will be different than when all message nodes are aligned in parallel. But if a combination of tasks and messages are aligned in parallel, the slot length will be determined by the entity with larger time requirement. Due to unequal time requirements, holes can be created in the schedule as shown in Fig. 2.7.

If the finish time of the last node is greater than the deadline, then this set of tasks and messages cannot be feasibly scheduled. Please note that the objective in step 2 is not to create a minimum makespan schedule. Rather it is to create a slotted schedule such that entities are aligned in parallel. This approach is so designed because it makes slack allocation in step 3 computationally easier to perform as described in the next section. For the mixed graph shown in Fig. 2.6, the initial schedule can be constructed as shown in Fig. 2.7. The tasks of the leaf nodes are all scheduled in parallel in the beginning of the schedule. They are removed in Fig. 2.7 for the sake of clarity.

2.6.3 Slack Reclamation

Any time slot (slack) not occupied in the initial schedule can be used by an entity to lower its performance level. Additional slack allocation to an entity may result in a lower performance level thus reducing the energy consumption of the network. This implies altering the previously assigned start and finish times of the scheduled entities. Considering the conflicts present, altering the start and finish times of a single entity might lead to a domino effect, where in the worst case, the schedule of all the successive entities needs to be altered by varying amounts. Due to this, allocating slack to any single entity would result in modifying the start and finish times of O(V_M!) entities, where V_M is the total number of nodes (tasks and messages) in the graph. This becomes computationally very expensive with increasing network size. The solution to this problem demands finding a subset of entities which will yield the highest rewards in terms of energy savings, while leading to as simple conflict management as possible. In order to do that, slack distribution needs to be performed in two steps:

1. All Rows Slack Reclamation
2. Scattered Hole Reclamation

2.6.3.1 All Rows Slack Reclamation

The energy saving capability of each entity can be calculated as the reduction in energy consumption per unit of additional slack allocation. We use the energy gain metric to select the entity that gives the highest energy savings for each additional unit of time allotted to it. We use a matrix like data structure to find this set of entities. We transform the schedule constructed in step 2 into an Energy gain matrix $\mathcal{A}$. Each column c of the matrix corresponds to the set of entities scheduled in parallel and each row r corresponds to the set of entities scheduled serially. The entry, $\mathcal{A}_{rc}$ in the matrix is the energy gain value $\mathcal{E}_{rc}$ of the corresponding entity v_r in the schedule. This value is subject to change once the entity has been scaled down. We configure a greedy algorithm that searches for a column of the matrix $\mathcal{A}$, such that the sum of energy gain values of entities in that column are the highest among all columns. For a

column c, the sum of energy gain values of entities scheduled in parallel is denoted as Sum_c. Allotting a portion of the slack to this column, simplifies conflict management as the schedules of all the successive columns can be altered by this fixed amount of slack. Algorithm 2 presents the pseudo code for slack reclamation. We define reclaimable slack as the difference between the schedule finish time and the deadline. The all rows slack reclamation algorithm is capable of recognizing and exploiting slack that is present across all the rows of the schedule. In the next section, we discuss how to exploit slack that is scattered as holes in the schedule.

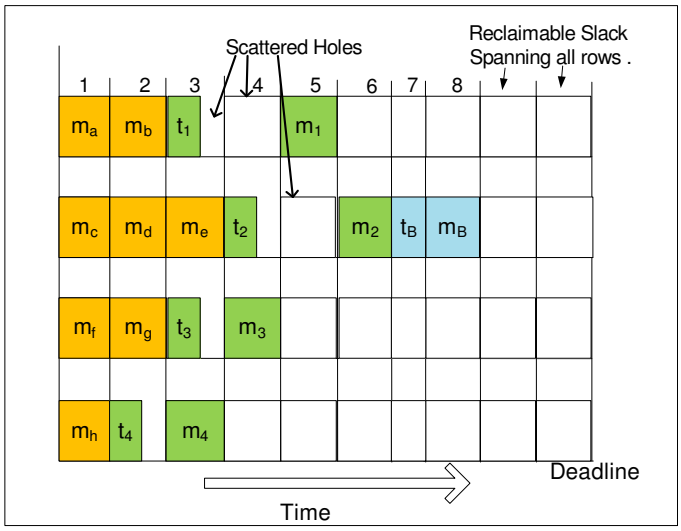

Figure 2.7: Example schedule showing reclaimable slack and scattered holes

The matrix $\mathcal{A}$ can be written as,

$$\begin{pmatrix}
\mathcal{E}_{t_a} & \mathcal{E}_{m_a} & \mathcal{E}_{m_b} & \mathcal{E}_{t_1} & - & \mathcal{E}_{m_1} & - & - & - \\
\mathcal{E}_{t_b} & - & - & - & - & - & - & - & - \\
\mathcal{E}_{t_c} & \mathcal{E}_{m_c} & \mathcal{E}_{m_d} & \mathcal{E}_{m_e} & \mathcal{E}_{t_2} & - & \mathcal{E}_{m_2} & \mathcal{E}_{t_B} & \mathcal{E}_{m_B} \\
\mathcal{E}_{t_d} & - & - & - & - & - & - & - & - \\
\mathcal{E}_{t_e} & - & - & - & - & - & - & - & - \\
\mathcal{E}_{t_f} & \mathcal{E}_{m_f} & \mathcal{E}_{m_g} & \mathcal{E}_{t_3} & \mathcal{E}_{m_3} & - & - & - & - \\
\mathcal{E}_{t_g} & - & - & - & - & - & - & - & - \\
\mathcal{E}_{t_h} & \mathcal{E}_{m_h} & \mathcal{E}_{t_4} & \mathcal{E}_{m_4} & - & - & - & - & -
\end{pmatrix}$$

2.6.3.2 Scattered Hole Reclamation

At the end of the all rows slack reclamation phase, all the reclaimable slack is utilized and the schedule makespan equals the deadline. However, slack could still be present as holes in

Input: Energy-unaware schedule, Sch_u
Output: Energy-aware schedule, Sch_e
1. Calculate Reclaimable slack,
   R = Deadline- Schedule makespan
2. For each entity v_a in Sch_u, calculate Energy-gain
   as, $\mathcal{E}_a = \frac{\mathcal{E}_a^k - \mathcal{E}_a^{k-1}}{t_a^{k-1} - t_a^k}$
   where $\mathcal{E}_a^k$ = energy consumption of entity a at level
   k and t_a^k = time consumption of entity a at level k.
3. If $\mathcal{E}_a < 0$ then $\mathcal{E}_a = 0$.
4. Construct an m x n matrix, $\mathcal{A}$ of energy gain values.
5. For every column n, calculate $sum_n = \sum_{r=1}^{m} \mathcal{E}_{rn}$
6. **while** $R \geq 0$ **do**
   Find column c having the maximum sum_c,
   calculate $shift = \max\{(t_{1c}^{k-1} - t_{1c}^k), \ldots (t_{mc}^{k-1} - t_{mc}^k)\}$
   if $shift \leq R$ **then**
   k= k-1 for all entities in the column c.
   Modify the finish time.
   R = R- shift.
   end
end
/* end of All rows slack reclamation phase.*/
7. Let S be the set of all entities sorted in
   non-increasing order of their Energy gain.
8.**while** $S \neq NULL$ **do**
   Pick an entity i from S.
   if $(St_{Successor(i)} - Ft_i) \geq (t_i^{k-1} - t_i^k)$ **then**
   if *i does not violate any constraints with neighboring nodes* **then**
   reduce performance level.
   Change the finish time.
   end
   end
   Remove i from S.
end
/* end of Scattered slack reclamation phase.*/

Algorithm 2: Two phase slack reclamation

the schedule as shown in Fig. 2.7. Variation in the execution times of entities and variation in the energy gains of different entities explain the presence of this slack. Such slack should be allocated to an entity which can use this slack without violating any constraint. The way initial schedule is constructed, each node is assigned the smallest possible start time. Further, at the termination of all rows slack reclamation phase, the schedule makespan is equal to the deadline leaving no room for shifting of a set of serially scheduled entities. This means that this type of slack could be best utilized by the individual entity adjacent to the hole such that if this entity did not interfere with any neighboring node in the beginning, then it must not interfere with it once the slack is allotted to it. For every such slack allocation to any entity, the start and finish times of the neighboring nodes (in adjacent columns) in the schedule are

checked for constraint violations.

2.6.4 Computational Complexity

We assume that there are V nodes and E edges in the network graph to begin with. Upon conversion to mixed graph, there will be $V_M = (V * 2)$ nodes. In step 1, finding the interference degree of a node requires searching the entire vertex set V_M to find nodes that satisfy Eq. 2.9 which can be done in time $O(V_M^2)$. In the initial schedule construction algorithm, sorting the nodes in non increasing order of priority takes $O(V_M\ logV_M)$ time (step 3). Again sorting in non increasing order of degree takes $O(V_M\ logV_M)$ time . Allocation of start time and finish time to each node in step 4 can be done in $O(V_M^2)$ time. Thus the complexity of initial schedule construction phase is $O(V_M\ logV_M + V_M^2)$.

At the end of the initial schedule construction, the availability of slack is known, at which point slack distribution can be performed. In the worst case, all the performance levels of all the nodes in the schedule might be checked. The computational complexity of the slack reclamation algorithm can then be derived as $O(V_M^2(\lambda_t + \lambda_m))$, where λ_t is the number of available discrete frequencies for DVS and λ_m is the number of available discrete modulation levels for DMS. The overall complexity of the proposed heuristic is thus derived as $O(V_M\ logV_M + V_M^2 + V_M^2(\lambda_t + \lambda_m))$.

2.7 Performance Evaluation

We perform extensive simulations to access the performance of our proposed algorithm IEJSH. Due to the absence of any existing related work with the same set of objective function and constraints, we compare our results with the following three: optimal results obtained by the MILP; a scheme *NoCheck* which does not take into consideration the nature of DMS; and a scheme *NoScaling* which constructs an initial feasible schedule according to Algorithm 1, but does not perform either DVS or DMS. The performance metric of interest is the *normalized energy consumption* of the network. The total energy consumption of a network is calculated using the Eq. 2.14. This energy consumption is then normalized with respect to the energy

consumption of the *NoScaling* scheme which does not scale entities at all. Each point in the graph is the average of 20 simulation runs.

Widely available motes such as iMote-2 use the DVS technology, but currently there are no motes available which implement the DMS capability. Hence, we build a custom Java simulator to simulate the different scenarios. We compare the results of our scheme with the optimal results obtained through the MILP formulation. We used the ILOG CPLEX 12.2 software [34] to solve the MILP. We generate networks of varying sizes. For each network, we randomly assign location to the nodes in a $100m$ x $100m$ area. The range of the sensor node is taken as $20m$. The complexity of the MILP can be estimated by the size of the input problem. For a 50 node network, there were 4000 binary variables, 400 continuous variables and 20500 constraints.

2.7.1 Simulation Parameters

For each simulation, we use the following parameters:

- *Computation Subsystem:* The CPU is assumed to be the Intel Xscale PXA27x CPU which is used on iMote-2 sensor node. Task Cycles, C= 5000, frequency f can take values from the set F= {624MHz, 520MHz, 416MHz, 312MHz, 208MHz, 104MHz} and the associated power consumption is P= {925mW, 747mW, 570mW, 390mW, 279mW, 116mW} [38].

- *Communication Subsystem:* Data packet size, L = 2000 bits, N_f = 10dB, σ^2 = -174dBm/Hz, δ = 0.1, P_b = 10^{-3}, W= 3MHz. Modulation level b could take values from the set, b = {2, 4, 6, 8, 10}. Given a transceiver structure, certain components can be evaluated to be constant like circuit power, P_c = 0.2505 mW, and transient energy, E_{tr} = 0.5*10^{-6} J [26].

Next, we present the effect of variation of different parameters on the network energy consumption. Specifically, we vary the average internode distance, d_{avg}; packet size per message, L; computation cycles per task, C; number of nodes in the network; and the slack factor, sf.

Slack factor is defined as $\frac{D-Sch_f}{Sch_f}$, where D is the deadline and Sch_f is the schedule makespan at the end of the initial schedule construction phase (step 2).

2.7.2 Effect of slack variation

Fig. 2.8 and Fig. 2.9 show the effect of increasing slack on the total energy consumption of the network comprising of 50 and 100 nodes respectively. One could think that more slack should result in more energy savings. But as shown in Fig. 2.5, at lower transmitter-receiver distances, the circuit consumption energy becomes a dominant factor. Further, at lower modulation levels, when the transmitter receiver circuit needs to be *on* for a longer duration, the circuit consumption energy overshadows any energy savings acquired due to

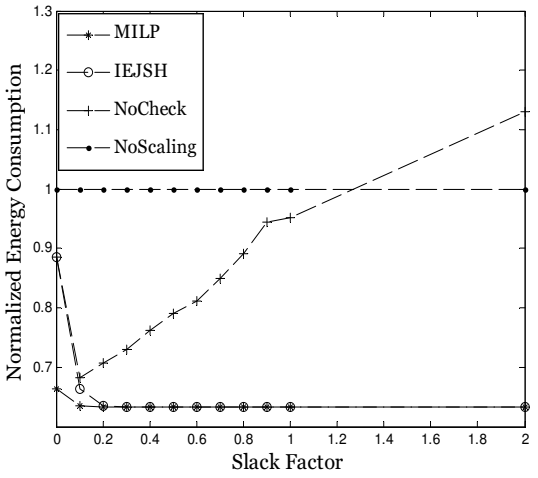

Figure 2.8: Variation in slack factor, Nodes = 50

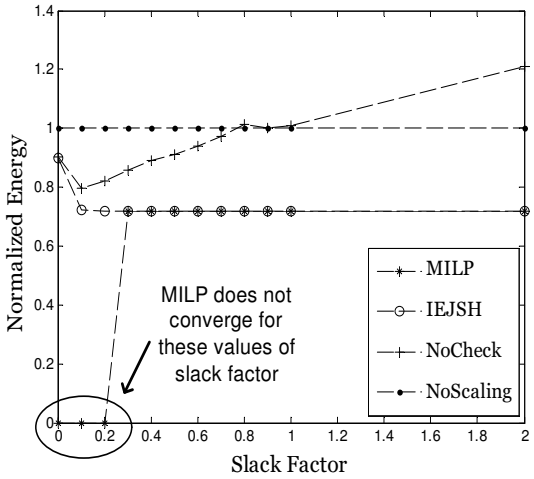

Figure 2.9: Variation in slack factor, Nodes = 100

modulation scaling. Thus at smaller distances, the lower modulation levels do not necessarily give better energy savings. This means that slack should only be allocated as long as there is positive energy savings.

Notice that the energy consumption of $NoCheck$ scheme keeps increasing and crosses that of $NoScaling$ (at sf > 1). This is because $NoCheck$ keeps reducing the modulation levels with increasing slack factor without ensuring if this slack allocation results in positive energy savings. The lower modulation levels result in overall higher energy consumption.

On the other hand, $IEJSH$ ensures that the successive slack allocation should result in

energy savings. $MILP$ and $IEJSH$ show better energy consumption than $NoCheck$ and $NoScaling$. The energy consumption of $MILP$ and $IEJSH$ reduces in the beginning but becomes constant after $sf = 0.4$. In these cases the optimal modulation level is not the smallest modulation level, instead it is the level after which there can be no positive energy savings with modulation scaling. At values of slack factor (≥ 0.2) $IEJSH$ gives optimal results. However, network energy optimization using $MILP$ is computationally expensive. For networks with 100 nodes, $IEJSH$ takes approximately 3-5 seconds, while the average running time of the MILP-based approach ranges from 0.34 sec ($sf = 1.0$) to 3940 sec ($sf = 0.3$) on a AuthenticAMD machine with a AMD Opteron 250 2.4GHz CPU. This is a significant reduction in the running time, while still obtaining near optimal solutions (0.12% more than the optimal solution for sf= 0.3 and exactly same values for $sf = 1.0$). For cases where the reclaimable slack is very less ($sf \leq 0.2$), the MILP does not converge on any solution (even after 70 hours) as shown in Fig. 2.9.

2.7.3 Effect of variation in packet size

Fig. 2.10 serves to show the pronounced effect of circuit consumption energy which becomes comparable to the transmission energy with increasing packet size and low transmitter-receiver distance. The average transmitter receiver distance is 10m. With increasing packet sizes the transmitter and receiver circuit needs to be *on* for longer time, thus increasing the circuit consumption energy. This effect becomes more dominant at low modulation levels. $NoCheck$ scheme which kept lowering the modulation level resulted in higher energy consumption than if it had not opted for any scaling (similar to $NoScaling$). Higher energy consumption by low modulation levels is also shown by the crossover in the energy consumption performance of $NoCheck$ and $NoScaling$ scheme at packet sizes ($\geq$ 2500bits). Higher modulation is capable of more energy savings as compared to the low modulation levels. This is shown by reduced energy consumption of $IEJSH$ and $MILP$ because they stop scaling once it results in negative energy savings. $IEJSH$ performs closely as compared to the $MILP$. At L= 1000 and 5000, $IEJSH$, respectively, incurs as small as 0.29% and as large as 7.5% more energy than the

$MILP$ as shown in Fig. 2.10. Please see Fig. 2.5 for better understanding of the relationship between energy components and transmitter-receiver distance.

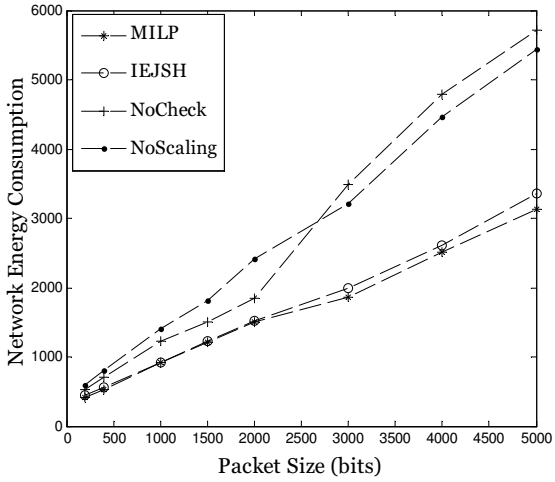

Figure 2.10: Variation in packet size

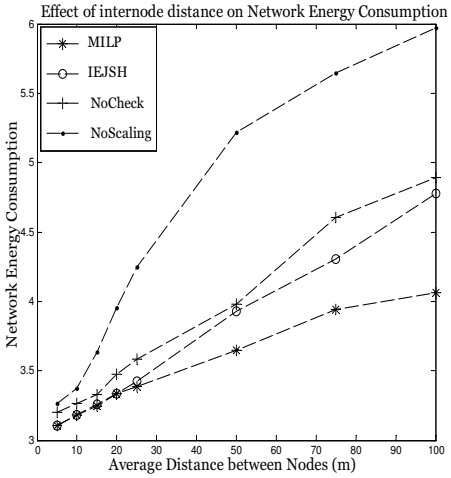

Figure 2.11: Variation in inter-node distance

2.7.4 Effect of variation in average distance between nodes

Fig. 2.11 shows the effect of variation of average distance between nodes on the total energy consumption. As the distance between nodes increases, the effect of interference reduces. Thus more and more nodes could be scheduled in parallel, which leads to a huge amount of slack. The transmission energy is an exponential function of the distance, hence increase in the transmitter-receiver distance leads to higher transmission energy consumption. This also means higher energy savings per step down of the modulation. This explains the drastic reduction in the energy consumption of our heuristic versus $NoScaling$, where no scaling is performed. Note that the vertical axis represents energy consumption in logarithmic scale for the sake of clear representation. $MILP$ and $IEJSH$ give exactly similar results at lower distances. This is because at such lower distances the higher modulation levels are the best modulation level. At higher distances, the results of $IEJSH$ match closer to $NoCheck$ as compared to $MILP$. This is obvious since transmitter-receiver distance is the major factor which differentiates between $IEJSH$ and $NoCheck$.

2.7.5 Evaluation of Initial Scheduling Phase

We evaluate the performance of the initial schedule construction phase (step 2) of our proposed algorithm with respect to the schedule makespan. We compare it with the schedule makespan resulting from an optimal initial scheduling phase. Our proposed slotted approach for step 2 creates an initial schedule with greater makespan than the optimal. The factors that play a crucial role are the different sizes of tasks and messages. We recognise that step 2 may not be able to construct a feasible schedule at times when deadlines are very stringent. However, this design serves to make slack allocation in phase 3 computationally simpler.

In these results, please note that while the proposed algorithm creates an initial schedule with greater makespan, this design serves to make slack allocation in phase 3 computationally simpler.

Fig. 2.12 shows the performance of initial scheduling phase with respect to varying topology. The cluster size is varied from 2 nodes to 10 nodes in each cluster while keeping the network size same at 50 nodes. We notice that varying topology does not affect the performance of our algorithm as shown by the constant gap between the performance of proposed approach and optimal solution.

The main reason behind the difference between the performance of initial scheduling by proposed approach and the optimal scheduling is the inequality in the size of task and message. Fig. 2.13 shows the performance of step 2 with respect to varying ratio between the computation time and communication time. We keep the message size constant and vary the computation cycles thus increasing the computation time required by each node. Notice that when the ratio between task and message size is close to 1, there is very less gap between the performance of our approach and the optimal. This gap keeps increasing with increasing inequality in task and message sizes. as shown in Fig. 2.13. However, these results of step 2 are given as input to step 3 where slack allocation is performed. When we compare the results at the end of step 3 as shown in Fig. 2.14, we see that there is a constant gap between the energy consumption between proposed algorithm and optimal solution even with increasing computation load.

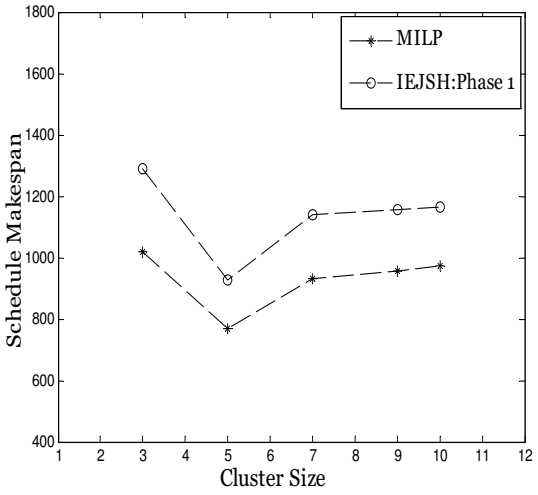

Figure 2.12: Effect of varying topology on schedule makespan

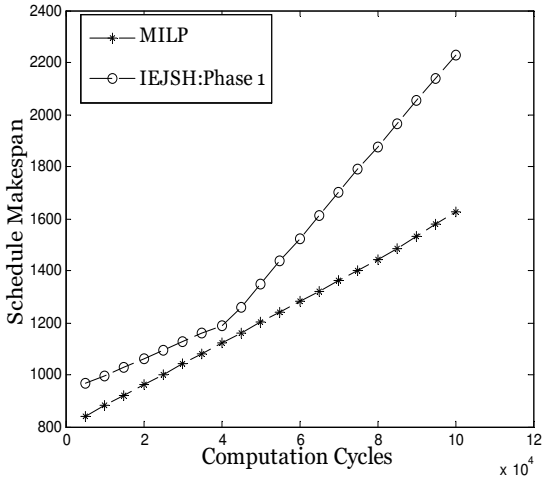

Figure 2.13: Effect of different task-message sizes on schedule makespan

2.7.6 Effect of variation in computation workload

Fig. 2.14 shows the effect of variation in computation load on the total energy consumption. Due to difference in the task and message sizes, holes are created in the schedule. This slack present as holes can be left unclaimed by messages due to high degree of interference amongst them. At this point, tasks can easily reclaim that slack (though often for a smaller energy gain) given that tasks do not have any interference constraints with other entities. Now, the larger the difference in task and message time, the more slack exists as holes and hence more energy savings can be incurred when tasks scale down. But with increasing computation cycles, tasks occupy more time thus reducing the slack size and hence lesser opportunities for tasks to scale down. Thus energy savings reduce with increasing computation cycles. At all times, $MILP$ being capable of an exhaustive search utilizes the slack in a better manner than our heuristic. The change in computation workload behaves differently than change in communication workload (as shown in Fig. 2.10). Notice that in this case $NoCheck$ consistently performs better than $NoScaling$. This is because tasks have monotonically decreasing energy consumption with decreasing frequency levels.

2.7.7 Effect of Network size

Fig. 2.15 shows the effect of network size on the energy consumption. The static slack provided as input is equal to 0.2. The Fig. 2.15 shows that with increasing network size, $IEJSH$ continues to yield the best performance as compared to the other schemes. For

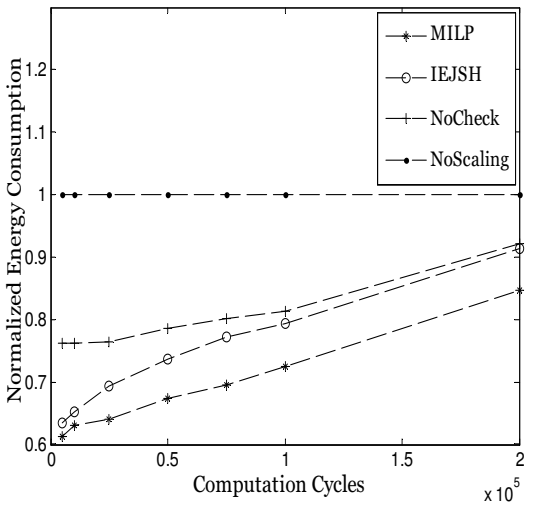

Figure 2.14: Variation in computation load

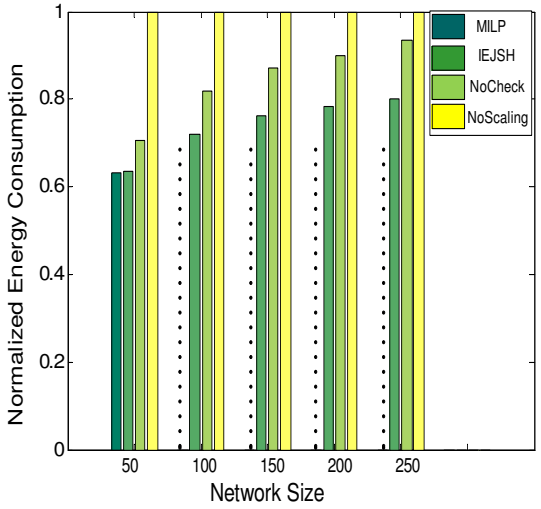

Figure 2.15: Variation in Network Size

networks with 50 nodes, $IEJSH$ yields similar results (0.16% more than the optimal result). Also, for networks consisting of more than 50 nodes, the $MILP$ does not converge on any result (even after 70 hours) on a AuthenticAMD machine with a AMD Opteron 250 2.4GHz CPU. Absence of any result by $MILP$ is shown by dotted lines. Thus, it is evident that with increasing network size, it becomes computationally very expensive and even infeasible to find the optimal result using $MILP$.

2.7.8 Comparison of different algorithms

Comparison between the different schemes with respect to computation time required in scheduling is shown in Table 2.2. The energy consumption by each scheme for computing a schedule is proportional to the total computation time required by the scheme. Thus higher the computation time, higher is the energy consumption by the scheme. We enlist the computation time required (which is a measure of computation energy required for scheduling) and also the

normalized energy consumption of the resulting scheduled network. Notice that computation time becomes a dominant factor while solving the problem through $MILP$. At high network size (= 100 nodes) and strict deadlines (slack factor ≤ 0.2), MILP does not converge even after 70 hours. The performance of all other schemes remains similar with respect to scheduling time (of the order of 1-2 sec).

Table 2.2: Computation time and energy comparison

Number of Nodes	Slack Factor	MILP	IEJSH Scheme	NoCheck Scheme	NoScaling Scheme
25	0.1	(0.543, 343s)	(0.544, 1s)	(0.566, 1s)	(1, 1s)
	0.2	(0.541, 127s)	(0.542, 1s)	(0.582, 1s)	(1, 1s)
	1	(0.540, 0.01s)	(0.540, 1s)	(0.857, 1s)	(1, 1s)
50	0.1	(0.634, 2454s)	(0.664, 2s)	(0.682, 2s)	(1, 2s)
	0.2	(0.633, 348s)	(0.634, 2s)	(0.707, 2s)	(1, 2s)
	1	(0.632,0.24s)	(0.632, 2s)	(0.951, 2s)	(1, 2s)
100	0.1	(–, 70hr)	(0.720, 2s)	(0.796, 2s)	(1, 2s)
	0.2	(–, 70hr)	(0.719, 2s)	(0.820, 2s)	(1, 2s)
	1	(0.718, 2.3s)	(0.718, 2s)	(1.01, 2s)	(1, 2s)

2.8 Conclusion

In this chapter we formulated the problem of joint scheduling of tasks and messages in the presence of precedence, interference and deadline constraints. We provided a proof of the hardness of the stated problem. We presented a three phase polynomial time heuristic to perform scheduling with the objective of energy minimization in the presence of these constraints. We efficiently performed slack allocation considering that in dense deployments of WSN with small transmitter receiver distances, DMS does not monotonically reduce the energy consumption [26]. We evaluated the performance of the proposed algorithm for a variety of scenarios and our results show that the energy savings obtained by the proposed algorithm competes closely with that of the MILP solution.

CHAPTER 3 Online scheduling: Energy-aware Adaptive MAC Protocols

3.1 Summary

In this chapter, online scheduling algorithms that can utilize resources left unused at actual runtime conditions to minimize energy consumption in Real-time Wireless Sensor Networks (WSNs) are proposed. Adaptive algorithms are developed which exploit the energy-latency tradeoff by utilizing runtime slack. The proposed algorithms work at node level such that each node can independently perform some action to reduce its energy consumption during its operational time. For this purpose, two widely used Medium Access Control (MAC) schemes are targeted, namely, Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). In both the schemes, different opportunities to reduce energy consumption are identified at the node level during the node's runtime.

With respect to TDMA systems, the proposed approach differs from existing research on accounts of lightweight slack generation method by exploiting temporal correlation and achieving energy minimization by trading generated slack for energy savings using DMS. Then, heuristics with varying complexities are presented to reduce energy consumption while preserving constraints.

For CSMA/CA systems, the absence of channel adaptation and load adaptation results in energy wastage due to retransmissions and low success rate. In order to address this, an Adaptive-CSMA/CA protocol which accounts for varying channel and load conditions at a node by influencing the selection of either low energy or low delay transmission option is presented. An energy-delay metric is devised that helps a node select the best message and

modulation to obtain joint reduction in energy and delay at the time of transmission.

The rest of the chapter is organized as follows: section 3.2 explains the proposed energy aware scheduling algorithms with respect to the TDMA system. Section 3.3 presents scheduling algorithms with respect to the CSMA/CA system. Conclusion is presented in section 3.4.

3.2 TDMA systems: Energy minimization by exploiting data redundancy

WSNs are used widely for monitoring applications where the main purpose is continuous data collection of environment parameters such as temperature, pressure, wind speed etc. Since sensor data streams are measurements of continuous physical phenomenon, spatial and temporal correlations within data streams are inherent. Such spatial and temporal correlation exhibited by WSNs can be exploited to reduce energy consumption.Thus the probability that the data sampled by a node is highly correlated or repetitious over time is quite high. Consider the sample data trace shown in Fig. 3.1 consisting of two days of temperature variation in a room [1]. The Analog to Digital Converter (ADC) readings correspond to the temperature sensor readings. The horizontal portions in the trace are circled to show temporally correlated readings. In scenarios like this, making a smart decision by not sending the same repeating sensor measurement could save a large amount of energy. We propose schemes that employ such smart decisions and reduce the energy consumption of the network.

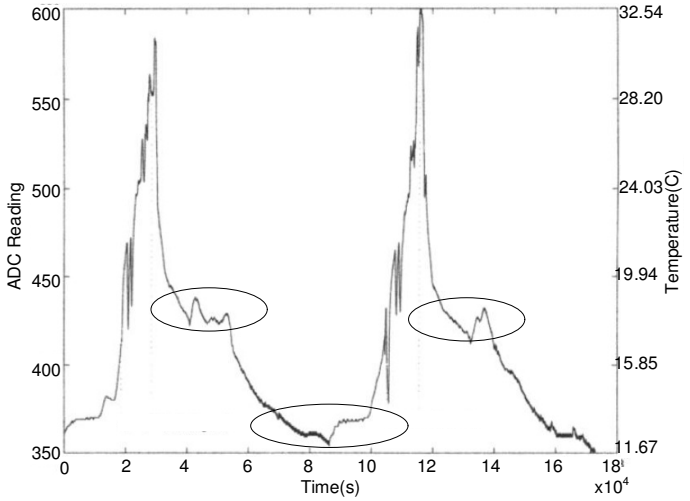

Figure 3.1: Sample trace of temperature sensor [1]

Previous research works attempt to exploit this correlation by way of aggregation or adaptive sampling. We propose determination of the data correlation at each node by way of local computation and propose avoiding transmission of significantly similar data. This can lead to unused time slots at runtime (dynamic slack) which can be traded off for energy savings. Our approach uses techniques namely Dynamic Voltage Scaling (DVS) and Dynamic Modulation Scaling (DMS), which utilize the slack generated dynamically to reduce energy consumption in a real-time environment. We propose heuristics with varying complexities for efficient slack management. We evaluate the performance of these heuristics by simulating diverse network conditions while incorporating different overheads. Our results show that the proposed heuristics can achieve energy savings up to 40% more than the baseline algorithms employing DVS and DMS, and, can achieve performance competitive with a Clairvoyant algorithm under network scenarios with high volume of redundant messages.

In the first part of this chapter, we propose a communication scheduling mechanism for a duty cycled data collection tree network. The main contributions of this chapter are as follows:

1. We propose a lightweight node level slack determination scheme by exploiting *approximate, bounded-loss* data collection in sensor networks. Each node independently decides whether the currently sampled data is significantly different from the previous reading. If the data is not significantly different, it does not need to be transmitted. This smarter way of not transmitting a repetitious data packet has two advantages: firstly, communication energy savings and secondly, the unused time slot (slack) can be used by other nodes to reduce their communication energy.

2. We propose smart slack distribution schemes. The proposed schemes detail the effective utilization of the resource (slack) left unused by nodes having redundant message. The schemes work in a distributed manner such that control overhead is minimized while the energy savings can be increased.

In this chapter we show that our approach based on smart slack determination and distribution, is capable of significant reduction in energy consumption as compared to the existing schemes.

We concentrate on the communication subsystem since it dispenses the maximum share of energy.

3.2.1 Related Work

A taxonomy of approaches for energy saving in sensor networks can be found in [39] and [40]. Numerous methods have been proposed to lower energy consumption of a WSN by suppressing redundant information in continuous data collection applications. These energy efficient data acquisition methods can be categorized into adaptive sampling schemes [1], [41], [14], [15] and [42]; model based schemes [43] and [44]; and data aggregation schemes [45], [46].

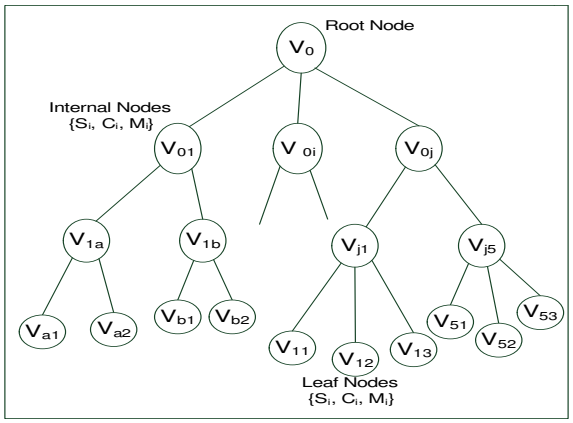

Figure 3.2: System Model

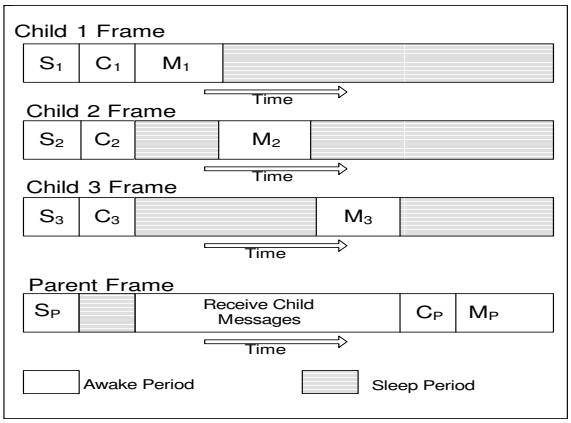

Figure 3.3: Schedule of task set at each node

Adaptive sampling schemes [1], [41], [14], [15] and [42] propose adaptive adjustment in the sampling rate of sensors according to the statistical features of the environment being monitored. In [41] and [14], authors build a spatio temporal correlation aware framework and perform adaptive and coordinated sampling in order to minimize energy. In [15] and [42], authors describe an adaptive sensor sampling scheme where nodes change their sampling frequencies autonomously based on time-series forecasting, so as to reduce energy consumption. However, these methods are energy efficient only in highly correlated scenarios. This is because these schemes call for regular control frame exchange amongst participating leaf nodes and the cluster head, resulting in non-trivial overhead due to control message exchange in mildly correlated applications.

Approximate data collection methods [43] and citeApproxData create models of the sensor data and the future data is then predicted based on these models at the sink thus removing the need for regular transmission by the source nodes. In [43], authors propose approximate data collection with bounded loss through the use of replicated dynamic probabilistic models. The basic idea is to maintain a pair of dynamic probabilistic models over the sensor network attributes, with one copy distributed in the sensor network and the other at a base station. The sensor nodes always possess the ground truth, and proactively route the data back toward the base station only when they sense anomalous data. [44] proposes an estimation model and proves rated error bounds of data collection using this model. Although these methods leverage the spatio-temporal correlation, they require complex estimation model generation which has to be application specific and the estimation models need regular fine tuning and updating. In [47], such associated overheads due to prediction have been categorised and the tradeoff is examined.

Data aggregation is proposed in [45], [46] to generate a concise report of the data. In [46], a combined approach to data aggregation and energy reduction using Dynamic Modulation Scaling [2] is proposed. However in these schemes, data aggregation happens at cluster heads. The repetitious, redundant data still needs to be communicated by each sensor to the cluster head before being termed as redundant. This leaf–aggregator node communication expends a large amount of energy over extended time periods.

Our solution is tailored for continuous data collection applications of WSN, where temporal correlation in sensor data can be exploited in a lightweight manner to generate runtime slack which can then be traded off to minimize energy consumption. We use two energy minimization schemes: Dynamic Modulation Scaling (DMS) [2] and Dynamic Voltage Scaling (DVS). These are techniques used to scale the message modulation levels (DMS) or processor frequency and voltage levels (DVS) such that the energy consumption of the messages (DMS) or tasks (DVS) is reduced at the expense of increased performance time. [23] and [24] propose making use of energy latency trade-off and the combined usage of DVS and DMS in order to minimize the energy consumption of networked systems. These works present centralized scheduling

algorithms that can tradeoff static slack to obtain energy savings. In [24], energy aware data acquisition approach specific to WSN is presented. However, authors do not take into account any spatio-temporal correlation in sensor data. In contrast, we propose algorithms for energy aware runtime scheduling and combine the benefits of data correlation and slack reclamation through DMS. Finally, we must note that our approach can also be used in conjunction with other techniques such as [43], for example, to combine the advantages of energy minimization techniques and advanced spatio-temporal correlation models.

3.2.2 System Model

We consider a WSN consisting of n nodes. The network is generally configured in the form of a unidirectional tree as shown in Fig. 3.2, for the purpose of data collection from several leaf nodes to the root node.

- A leaf node i performs a set of periodic tasks, $T_i = \{ S_i, C_i, M_i\}$, where S_i is the sensing task, C_i is the computation task (message redundancy determination, data processing etc.) and M_i is the communication task (message transmission to its parent).
- The parent node aggregates the message received from its child nodes and transmits it to the higher level node. Root node performs data analysis of the collected data.
- At node v_{ij}, the computation task is denoted by C_{ij} having CC_{ij} computation cycles and the outgoing message is denoted as M_{ij} with L_{ij} bits.

The scheduling of tasks and messages follows several constraints as enumerated:

- *Precedence constraints:* Each node performs periodic sensing and computation task, at the completion of which it can begin its communication. Similarly each parent node starts its local computation only after receiving all the messages from its child nodes. A parent node buffers the messages from its child nodes before performing data aggregation. Fig. 3.3 shows the sequence of steps for a cluster with a *Parent* node and three *Child* nodes.

- *End-to-end latency constraint:* Each message that is generated at a leaf node should reach the root node within a specified deadline, D.

- *Interference constraints:* For channel access, we assume that each node is allotted a time slot for communication with its cluster head. The nodes in a cluster receive exclusive access to the wireless channel since the cluster head can only receive from one transmitter at a time. Nodes in two different clusters can have shared access to the channel if they do not interfere with each other. We follow the protocol model [35] for interference modeling in wireless networks, according to which, if a node v_i transmits to a node v_j, then this transmission is successfully received by node v_j if

$$\|v_j - v_k\| \; > \; (1+\delta)\, \|v_j - v_i\| \tag{3.1}$$

  for any node v_k simultaneously transmitting to any node v_m at the same time using the same channel. Using this model, consider two adjacent clusters with cluster heads denoted as i and j. Let d_i denote the maximum distance between i and any of its child nodes and m_i denote the minimum distance between i and any of the child nodes in the adjacent cluster. Now if conditions in Eq. 3.2 are satisfied then the simultaneous transmission by any two nodes in these adjacent clusters do not interfere with each other.

$$m_i > (1+\delta)\, d_i \;, \quad m_j > (1+\delta)\, d_j \tag{3.2}$$

Interference Model: In this chapter, we use the protocol model (a.k.a. disk graph model) as opposed to the more accurate but highly complex physical model (a.k.a. SINR model) [35]. Link scheduling is shown to be NP-complete in [48]. Given sufficient transmission power, fixed transmission range is considered unattainable due to the interference effect of simultaneously active unintended transmitters. Under the protocol model, the impact of interference from a transmitting node is binary whereas physical model treats interference at a receiving node as an aggregate noise from unwanted transmitters. Due to this simplification, protocol model suffers from the following *side effects*:

1. *Conservative Case:* For a transmission to be successful, the receiver should be outside of the interference range of *all* the non-intended transmitters. This can result in reduced

schedulability.

2. *Optimistic Case:* When a node falls outside the interference range of each non-intended transmitter, the protocol model assumes that there is no interference. This can result in lossy transmissions, as small interference from different non intended transmitters can aggregate and prevent achieving a minimum threshold necessary for successful receptions.

We emphasize that the following reasons help minimize the negative impact of interference on our proposed schemes:

1. The impact of interference from nodes in neighboring clusters is minimized by following condition in Eq. 3.2.

2. Nodes do not employ any distance based power control. This helps to define a common maximum transmission and interference range throughout the network.

3. We set the ratio of interference to transmission range to be equal to 2. Recent research [49] shows that it is possible to use the protocol model as a good simplification for the physical model when the interference range is set appropriately (the optimal ratio between maximum interference range and maximum transmitter range should be within [1.5, 2]). Our algorithms can be extended by following this work, however this is out of the scope of our current work.

Applications which benefit most from our schemes have strategic deployments with close-knit clusters and relatively large inter-cluster distance (larger than the maximum interference range). Consider application such as transmission line monitoring, where sensors on a transmission tower form a cluster and the neighboring cluster is on the neighboring transmission tower; or medical monitoring where a cluster of sensors is monitoring a patient in separate rooms etc. Other factors that can reduce the impact of interference are use of a different frequency by each cluster for intra-cluster communication or use of directional antennas instead of omnidirectional antennas.

Scheduling Model: We assume that an initial feasible schedule is constructed using known link scheduling schemes during the initialization phase after sensor deployment. If

static slack prevails after the construction of initial feasible schedule, the centralized offline scheduler can assign frequency levels to tasks and modulation levels to messages using the Gain based Static Scheduling (GSS) algorithm [23]. The resulting schedule is then provided as an input to our schemes. The communication pattern of the tree has a single repeating frame. Each frame consists of time slots for the nodes in a level to send messages to their cluster heads.

Communication model: The modulation scheme considered is Quadrature Amplitude Modulation (QAM). The channel is modeled as frequency-flat Rayleigh fading model. Let b denote the number of bits per modulation constellation symbol, $\frac{N_0}{2}$ denote the channel noise power spectral density and BER denote the bit error rate. $E_{transmit}$ is the necessary transmitted energy, d is the distance between the transmitter and the receiver and d_0 is a normalizing constant that depends on the wavelength.

We assume that the propagation loss follows a polynomial model and the power decays in the αth order of the distance. As a result, the total energy used in message transmission from node i to node j can be expressed as,

$$E_{ij_{transmit}} = (\frac{d_{ij}}{d_0})^{\alpha}.\frac{L_{ij}.(2^{b_{ij}}-1)}{6b_{ij}}.\frac{N_0}{BER} \tag{3.3}$$

In addition to the transmission energy, energy consumed by the transmitter (node i) and receiver (node j) circuitry, is given by,

$$E_{i_{circuit}} = \frac{L_{ij}\beta_i}{b_{ij}} \tag{3.4}$$

$$E_{j_{circuit}} = \frac{L_{ij}\beta_j}{b_{ij}} \tag{3.5}$$

where β_i and β_j are implementation-dependent constants. Thus, the total radio energy spent for a message transmission at modulation level b is given as,

$$E_{ij_{message}} = E_{ij_{transmit}} + E_{i_{circuit}} + E_{j_{circuit}} \tag{3.6}$$

W denotes the channel bandwidth in hertz. The corresponding channel transmission rate in bits per second can be calculated as Wb. We assume that the BER is low enough to provide

the necessary message reliabilities. The time taken to transmit an L-bit message in the shared wireless medium is calculated as $\frac{L}{Wb}$.

In DMS, additional slack allocation to a message results in a lower message modulation thus reducing the communication energy consumption of that node. Slack can be differentiated as static or dynamic.

Static slack: Static slack is the difference between the application specific deadline and the initial schedule makespan. The availability of static slack is known in advance. The existing GSS algorithm [23] computes an energy-aware schedule given the static slack in a centralized manner.

Problem Statement : The static slack allocation problem can be stated as: *Given a feasible schedule that satisfies the deadline and precedence constraints, allocate slack to different messages by varying their modulation levels such that the energy consumption is minimized while still preserving the feasibility of the schedule.* The problem is NP-hard and the complexity of the problem can be proved by the polynomial reduction of the Multiple Choice Knapsack Problem to the current problem [50].

Dynamic slack: Dynamic slack is generated at runtime due to runtime variations such as absence of messages due to spatial or temporal correlations, channel variations etc. Given the limited connectivity amongst sensor nodes distributed over a large physical area, the huge cost of communication, varying network conditions and varying node capabilities, making centralized scheduling decisions at runtime in order to allocate the dynamic slack is highly energy inefficient. In this chapter, we present three heuristic solutions to address the problem of runtime slack distribution in a distributed manner in order to achieve energy efficiency. *Our focus is on effective allocation of slack to relevant messages such that the slack generated at run time is utilized and the energy expenditure is minimized in a best possible manner while satisfying all the constraints, given a feasible static energy-aware schedule that does not violate any deadline or precedence constraints.*

3.2.3 Energy-aware Scheduling Algorithms

In this section we present a framework for effective energy minimization of a network by exploiting runtime variations. The two key functions required to be carried out are: (1) Slack Determination, and (2) Slack Distribution. By way of slack determination, the amount of slack that exists in the schedule in each frame is determined. Slack distribution determines the amount of slack to be allocated to a message and the resulting modulation level. For slack distribution, we propose three heuristics which take different approaches towards smart slack allocation amongst nodes in the network.

3.2.4 Dynamic Slack Determination

We define τ_s as the sensing period and R as the resolution required by the WSN application.

3.2.4.1 Slack Generation at a node:

Fig. 3.4 shows the comparison method at each node to identify whether a message is redundant or relevant. During the initialization phase, the sensors sample the parameter of interest and store this value as *Relevant* value. At each sensing period τ_s, a node wakes up and samples the environmental parameter of interest. The current sensor reading is compared with the stored *Relevant* value and if the difference between the two is less than the required resolution R, then the node terms the current value as *Redundant* and discards it. If the difference is greater than R, the current value becomes the new *Relevant* value. For the periodic instance where the current sampled value was computed to be *Redundant*, the node prepares only a header to be sent to the cluster head.

Header reception accomplishes two objectives. One is to notify the receiver that the transmitter node is actively sensing data and successfully transmitting to the intended receiver. Secondly, receipt of only a header is the notification that the sender node has a redundant value to report. Thus a successful header transmission may not transmit any useful data but it accomplishes the transfer of information that the sender node is alive, able to communicate and has a redundant message to report. Upon reception of a header, the cluster head uses

the last *Relevant* value reported by this child node for the purpose of aggregation. The radio energy consumption by a node sending a header, E_{header} can be calculated using Eq.(6).

The decision of redundancy or relevancy of a message depends upon R. Clearly, for a fine grained measurement, value of R needs to be small and for a coarse grained measurement this value can be large. Across different applications, value of R may differ, depending on the required granularity of data collection. The value of R can be changed if in the same application, a different level of sensitivity is required at a different point in time.

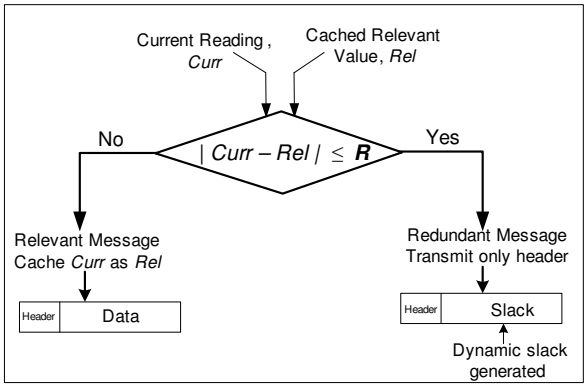

Figure 3.4: Dynamic slack generation at each node per sensing period

3.2.4.2 Slack identification by a peer node:

Now depending on the nature of the environment parameters and the resolution R, a varying number of sensor nodes will have redundant messages. Since these nodes transmit only a header, the time slot allotted for the message goes empty. The time left unused by a node is termed as *dynamic slack* since it is generated at runtime. This dynamic slack can be used by nodes in the same cluster which are scheduled later to scale down the modulation level of their messages, henceforth called as peer nodes. The decision of redundancy or relevancy of sampled data by a node need not be transmitted to peer nodes. In order to claim the dynamic slack, the peer nodes listen to the shared channel. An idle channel will signify that the predecessor node has finished sending a header. The peer nodes can then extend their assigned time slots. We characterize the overhead of listening energy in section 5.

3.2.5 Dynamic Slack Distribution

In order to optimize the energy savings, a decision needs to be taken regarding who should be assigned the runtime slack as different nodes give varying amount of energy savings per unit of time allotted to them. The decision of distribution of this extra slack cannot be done at the centralized scheduler all the time. In order to minimize energy consumption, this distribution of slack must be done in a distributed way.

At any instant in time, a subset of the nodes could have redundant message, we denote this subset of nodes as N. Likewise, the subset of nodes having relevant message is denoted as T. Also, each node expends some amount of energy in listening to the channel characterized by E_{listen}. The total energy consumption of the network can then be formulated as

$$E_{network} = \sum_{i=1}^{N} E_{i_{header}} + \sum_{i=1}^{T} E_{i_{message}} + \sum_{i=1}^{T+N} E_{i_{listen}} \tag{3.7}$$

We illustrate a simple example to aid understanding and comparison amongst all the schemes as described in later sections. Consider a simple network of 8 nodes as shown in Fig. 3.5 and consider two periodic instances over which our observation is based. The values on the edges represent the normalized distance between the corresponding nodes. In the energy-unaware schedule, each node performs a periodic computation task of $CC = 10^3$ computation cycles and periodic message transmission with packet size = 1024 bits. The values of different parameters is specified in Section 5. The energy-aware schedule that makes use of predefined static slack is shown in Fig. 3.6. The tasks and messages are at different modulation levels as decided by the algorithm, GSS proposed in [23]. For the sake of clarity, we omit the computation tasks. All nodes use the same channel for communication. Nodes communicating to the same cluster head follow a serial schedule while nodes communicating to different cluster heads can communicate in a parallel manner. The numbers on each time slot are the modulation levels at which these messages are modulated. Fig. 3.5 enumerates a table of typical energy and time values used in the examples. These values are obtained by using Eqs. 3.3-3.5 with values specified in Section 3.2.9. In the base case the given schedule incurs a total energy consumption of 10.88 mJ. Now consider 2 consecutive periodic instances of the schedule where nodes C and

G have redundant message. In the GSS algorithm since schedule decisions are made offline by a centralized scheduler and redundancy in sensing is not captured, the nodes C and G keep transmitting their sampled data. The total energy consumption is 10.88*2 = 21.76 mJ

Message Transmission Energy (b=10)	6.98 mJ
Header Transmission Energy (b=10)	0.982 mJ
Transmitter Circuit Energy per message	7.68 µJ
Receiver Circuit Energy per message	10.24 µJ
Listening Energy per channel sample	17.3 µJ
Message Transmission Time	102.4 µs
Header Transmission Time	14.4 µs

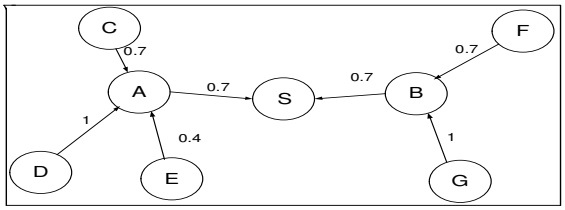

Figure 3.5: Sample Energy Values and Data Aggregation Tree

of energy as shown in Fig. 3.6.

In this chapter, we address the problem of energy minimization through runtime slack generation and distribution. However, in order for an algorithm to optimally allocate the dynamic slack generated at runtime amongst different messages, the algorithm needs an accurate global knowledge of the amount of slack generated networkwide as input. Since the accurate knowledge of future slack availability cannot be procured, an optimal online solution is not possible to attain. Hence, in order to utilize the available slack in the best possible energy efficient manner we present the following heuristic solutions. All the heuristic solutions follow the same method for slack generation at a node as described in section 3.2.4.1. At each node, slack generation requires one comparison between the current sensor value and the cached relevant value and hence takes $O(1)$ time. Modulation scaling can take $O(k)$ time in the worst case where k is the number of available modulation levels. The heuristics differ in their approach towards slack distribution. The rest of the section explains the slack distribution approach of the three heuristic algorithms namely: Basic, Weighted and Reshuffle, along with illustrative examples, pseudo codes and worst case complexity analysis for each of the schemes.

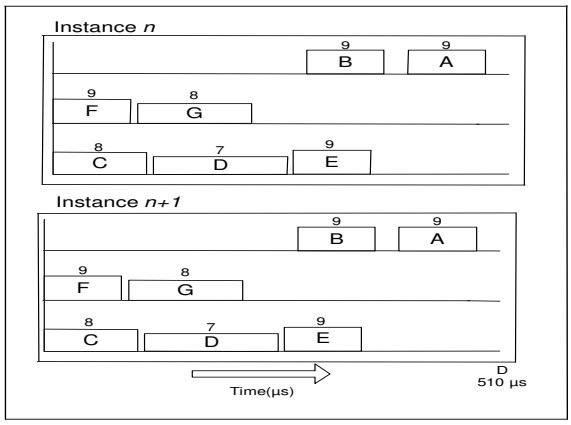

Figure 3.6: Baseline scheme,Energy Consumption = 21.76 mJ

3.2.5.1 Basic Scheme

This scheme implements a simple mechanism to utilize the dynamic slack generated. When a node has a redundant message and decides not to transmit it, the time slot goes empty. This empty slot can be fully claimed by the successor node if it can hear the predecessor node finishing early. Thus the main idea of the Basic scheme is: *The slack generated by a node is utilized solely by its immediate successor node in the schedule.*

However, the nodes are duty-cycled and wake up only at their assigned time slot. We assign an additional wake-up time to each node called $Start_{Listen}$ time when the node wakes up to see if the predecessor node has finished early. During network initialization, the schedule is constructed offline with the worst case assumption that all nodes will have a relevant message to send. The start time of any node will be greater than or equal to the finish time of its immediate peer predecessor node. Now at runtime, the predecessor node can either have a redundant message or a relevant message. Thus each node stores two values: one is the finish time of predecessor node when the predecessor node is relevant. This value will be known for each node when the input schedule is supplied. Second value is the finish time of predecessor node's header when the predecessor node has a redundant message. This time is termed as $Start_{Listen}$ and can be calculated as, $Start_{Listen}$ = scheduled start time of predecessor + header transmission time of predecessor.

Please note that if a node has a redundant message, it transmits a fixed sized header at

the highest modulation level. Thus irrespective of the modulation level assigned to a node, if redundant, all nodes transmit a fixed sized header at the highest modulation. Thus each node wakes up at $Start_{Listen}$ to sense the channel. If the channel is busy it means that the predecessor node has a relevant message and the node goes back to sleep to wake up at the time assigned for it to start transmission. Else if the channel is idle, it means the predecessor node has a redundant message and the node is free to reclaim this empty time slot thereby reducing its modulation level and thus reducing the communication energy. The nodes individually determine the best level for message modulation. At each node, redundancy determination will take O(1) time and modulation scaling can take O(k) time in the worst case where k is the number of available modulation levels. Since this happens serially in each cluster, the worst case running time of the basic scheme can be estimated to be O(Nk) where N is the size of the biggest cluster.

In the example discussed, node C and G on discovering that they have redundant messages, schedule only a header for the consecutive instance as shown by shaded slot. Node D hears node C finishing early and uses all the available slack to lower its modulation level as shown

```
Input: Energy Aware Schedule S_i
Output: Energy Aware and Redundancy aware Schedule S_o
Q : Set of all leaf nodes v_ij in S_i
while Q ≠ NULL do
    if v_ij has a redundant message then
        v_ij schedules only header
    end
    if Start_Listen(v_ij+1) ≥ ft(v_ij) then
        Dynamic slack available to v_ij+1
        dyn_sl = st(v_ij+1) -
        Start_Listen(v_ij+1)
        Let s be the time difference between v_ij+1's current modulation level and its next lower
        modulation level.
        while s ≤ dyn_sl do
            Reduce modulation level of v_ij+1 by one level.
            dyn_sl = dyn_sl - s
            update s
        end
    end
    Remove v_i from Q.
end
```

Algorithm 3: Pseudo code for basic scheme

by highlighted slot. The total energy consumption in this case is 19.26 mJ of energy as shown in Fig. 3.7. The pseudo code for Basic Scheme is shown below. In the rest of the chapter

we denote scheduled start time of a node v as $st(v)$, finish time as $ft(v)$ and listen time as $Start_{Listen}(v)$.

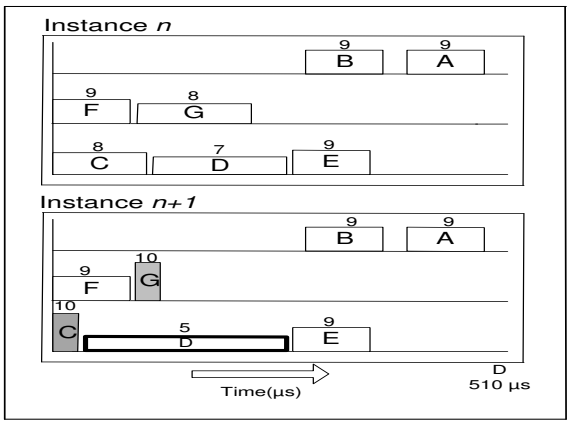

Figure 3.7: Basic Scheme, Energy Consumption = 19.26 mJ

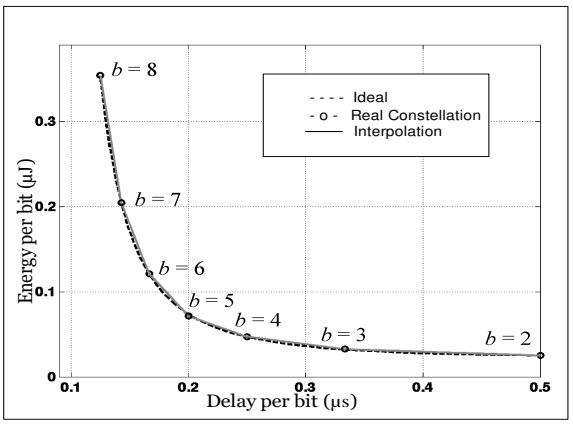

Figure 3.8: Energy Delay Curve [2]

3.2.5.2 Weighted Sharing Scheme

The Basic scheme proposes full usage of available slack by the immediate successor node. In a WSN with hundreds of nodes at different modulation levels, this scheme could be improved upon by noticing the trend in the energy saving capabilities of different nodes. In [2], authors investigate the nature of the energy-time trade-off curve. According to this curve, as shown in Fig. 3.8 all step downs contribute differently to the energy minimization process. Since the nature of the curve is convex and not linear, decreasing modulation level by equal amounts does not always give equal energy gains. This means that two nodes with similar parameters like node-cluster head distance, message size etc., but different modulation levels will give different energy gains when scaled down by the same number of modulation levels. For example, a node with a current modulation level of 8 when scaled to modulation level of 7 provides more energy gain as compared to a node (with similar parameters) with current modulation level of 5 when scaled to 4. This contribution can be effectively measured with the help of the *energy gain metric* as proposed in [23].

$$EG(k, k-1) = \frac{E_k - E_{k-1}}{\hat{t}_{k-1} - \hat{t}_k} \tag{3.8}$$

where E_k and $\hat{t}_k$, respectively, denote the energy consumption and time incurred by the node when operated at kth performance level. For each node e_j, which is currently assigned the kth performance level, EG(k, *k-1*) represents the energy reduction that would be obtained by reducing its performance level to *k-1* normalized with respect to the additional time incurred for operating it at performance level *k-1* instead of level k.

We use this metric to propose a scheme where peer nodes contend for the dynamic slack. The node offering the highest energy gain, gets the maximum share of the slack. The main idea of the Weighted sharing scheme is: *Slack generated by a node could be distributed amongst a set of successor nodes in a weighted manner such that the collective energy gain is higher.* Again this distribution needs to be done without the intervention of any centralized entity. When a node wakes up at $Start_{Listen}$ and realizes its predecessor node has a redundant message, it calculates its share of the slack. The share of slack allotted to this node is proportional to its current modulation level. A node v_{ij} decides its share of the slack as

$$share = \frac{b_{self}}{b_{total}} * dyn_{sl} \tag{3.9}$$

where, b_{total} is the sum of current modulation levels of v_{ij} and its peer successor nodes, b_{self}

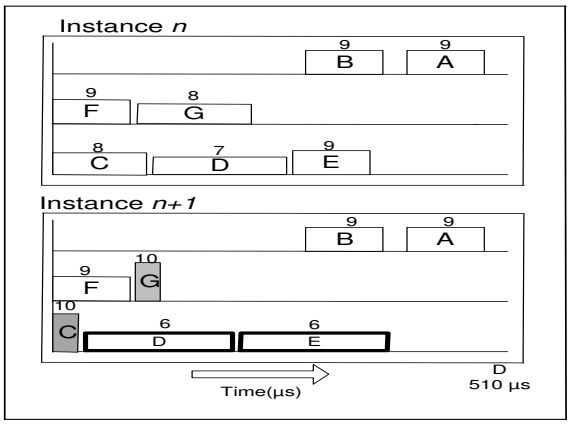

Figure 3.9: Weighted Sharing Scheme, Energy Consumption = 19.05 mJ

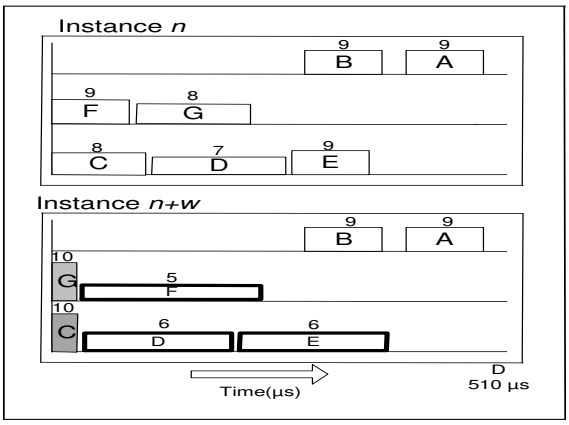

Figure 3.10: Reshuffle Scheme, Energy Consumption = 17.38 mJ

is the current modulation level of v_{ij} and dyn_{sl} is the dynamic slack calculated by v_{ij}. It decides the best modulation level to use this share of slack and the rest of the slack is left for the successive nodes to use. Since share calculation can be done in O(1) time by each node, the

worst case complexity of the weighted sharing scheme is the same as basic scheme. However, the amount of slack cannot be precalculated as basic scheme. This calls for the nodes to wake up more than once to listen for the possibility of available slack. In order to bound the listening overhead, nodes perform *strobed listening*, meaning the nodes wake up multiple times starting from $Start_{Listen}$ to the start of their assigned time slot. Strobed listening presents a node with the opportunity to capture slack being left for it to use while wasting lesser energy in listening.

Considering the example shown before, now the slack generated by C is used by both D and E, in a weighted sharing manner. The energy consumption here is 19.05 mJ.

Input: Energy Aware Schedule S_i
Output: Energy Aware and Redundancy aware Schedule S_o
Q : Set of all nodes v_{ij}in S_i
while $Q \neq NULL$ **do**
  if v_{ij} *has a redundant message* **then**
    v_{ij} schedules only header
  end
  If $Start_{Listen}(v_{ij+1}) \geq ft(v_{ij})$
  Dynamic slack available to v_{ij+1}
  $dyn_{sl} = st(v_{ij+1})$- $Start_{Listen}(v_{ij+1})$
  v_{ij+1} calculates *share*
  Let s be the time difference between v_{ij+1}'s current modulation level and its next lower modulation level.
  while $s \leq share$ **do**
    Reduce modulation level of v_{ij+1} by one level.
    $share = share$ - s
    update s
  end
  Remove v_{ij} from Q
end

Algorithm 4: Pseudo code for weighted sharing scheme

3.2.5.3 Reshuffle Scheme

The static allocation of time slots and exclusive sharing of the channel at the time of initial schedule construction can become a bottleneck in making use of the dynamic slack due to precedence constraints. Note that in Fig. 3.9 the slack generated by redundant node G remains unused since node F is scheduled before G. Given that high spatial and temporal correlation is inherent in sensor networks, the nodes in the area of redundancy can offer high energy savings if they are positioned in the schedule such that they can use the slack at the right time. Thus the main idea of the Reshuffle scheme is: *If some nodes continuously have*

redundant messages, their position in the schedule can be rearranged such that their successive peer nodes having relevant messages can utilize the slack generated by them.

This reshuffle is done in a locally centralized manner wherein a parent node makes decisions for its child nodes. Each parent node (cluster head) observes the trend in the data trace of each of its child node, for a pre-decided number of instants called an *observation window*. Each parent node can generate a model based on the past behaviour of its child nodes. The child nodes exhibiting redundancy majority of the time are selected to be at the beginning of the schedule.

The model used in order to decide the redundancy trend followed here is based on a simple majority rule. Each parent node keeps track of the number of times a child node exhibits redundancy in the current *observation window* and then schedules them in the decreasing order of redundancy count for the next *window*. The node exhibiting redundancy majority of the time in the past *window* is expected to behave in the same manner and is scheduled first in the next *window*.

The schedule reshuffle decisions are then broadcasted to the child nodes at the end of the observation window. Additional time slots are allotted for the schedule updates. Let w denote the size of the observation window. The complexity of linear prediction at the cluster head is estimated as $\mathrm{O}(Nw^2)$. This reshuffled schedule is followed for the next w instances and the leaf nodes follow weighted sharing scheme for slack utilization. Thus the complexity of the reshuffle algorithm can be estimated as $\mathrm{O}(Nw^2 + Nwk)$.

Coming back to our example, note that if node B decides to reshuffle the positions of node F and G, then slack left unused by G can be relocated such that it can be detected and utilized by F. Energy consumption in this case drops to 17.38 mJ as shown in Fig. 3.10.

The overhead incurred by the reshuffle algorithm is directly proportional to the number of times the schedule broadcast has to take place and hence inversely proportional to the size of the observation window. Alternatively, if the network is exhibiting fast changing characteristics and the window size is relatively large, it will not be able to capture the dynamic characteristic of the network and will make a reshuffle decision that is not correct. This suboptimal schedule

ends up causing more energy consumption. Thus calculating the size of the observation window is crucial in balancing the overhead incurred. Let us consider a trace of s instances and calculate overhead as a function of the window size. We formulate the total overhead as shown below,

Overhead = Energy wasted in broadcasting schedule + extra energy incurred due to a wrong decision.

$$Overhead = (\frac{s}{w}).e + p.s.q \tag{3.10}$$

where, s is the total number of instances, w is the window size, e is the energy consumption per broadcast in the network, p is the probability of error in linear predictor estimating the trend in changing network condition and q is the overhead encountered per wrong decision. This q is calculated as the average difference between the energy consumption in case a correct decision is made and the energy consumption when an imperfect decision is made.

Term e depends on the choice of parameters during the design of the network; in order to find the values of p and q, the network needs to be observed for some initial instances.

Input: Energy Aware Schedule S_i
Output: Energy Aware and Redundancy aware Schedule S_o
Observation Phase:
w = window size, ins = instance number
$\forall$ parent nodes v_i who have a leaf node v_{ij}
Q : Set of all child nodes v_{ij}
while $ins \leq w$ **do**
1 | Node v_i observes the finish time of v_{ij}
end
Enter Schedule Reshuffle Phase once every window
Use linear predictor to estimate the future values of each child node
Schedule Reshuffle Phase:
while $Q \neq NULL$ **do**
1 | find the node v_{ij} with highest redundancy count
2 | Schedule it at the earliest time.
end
$\forall$ leaf nodes v_{ij}
Follow Weighted Sharing Algorithm.

Algorithm 5: Pseudo code for reshuffle scheme

3.2.6 Enhanced Reshuffle Scheme

The reshuffle scheme makes decisions based on the redundancy count of the sensors in the previous observation window. This can be improved by exploiting the fact that physical environments frequently exhibit predictable stable states and strong attribute correlations that

can assist us in inferring the state of a sensor from its past and its surroundings. For example, outdoor temperatures typically follow consistent diurnal and seasonal patterns, and at any moment in time, are unlikely to vary greatly within a local region [43].

Our proposed Reshuffle scheme can be enhanced by utilizing these diurnal/seasonal data patterns to forecast the redundancy trend in future *observation window*. The data patterns can be recognised using one of several algorithms proposed for data pattern discovery and recognition in a time series [51]. At the end of every observation window, each parent node uses its knowledge of the future pattern for each child node. Using this future pattern, the parent node calculates the redundancy count which is the number of times this child node is going to have a redundant message in the next window. The nodes are then scheduled in a decreasing order of redundancy count.

Since sensor readings can always deviate from the modeled data pattern, Reshuffle Enhanced can also make mistakes. The data patterns can be updated with any outlier data. During the time that reshuffle decisions are broadcasted, the parent node can *condition in* the outlier data into the existing model using algorithms presented in [43]. In our work, the local node decides on the redundancy or relevancy of data and hence the *bounded loss approximation* guarantees are maintained irrespective of the scheme used and irrespective of the use of data patterns or not. So if at the local node, a currently sampled value is relevant, it will be scheduled by this node and the peer nodes will adjust their schedule online. Overhead in terms of energy and time required for computation of reshuffled schedule is added to the calculations.

3.2.7 Clairvoyant Scheme

We compare the performance of the proposed heuristics with a Clairvoyant scheme. Since an optimal online algorithm is not possible in practice, we emulate the Clairvoyant scheme like an *optimal online* algorithm. The Clairvoyant scheme has knowledge about the future redundancy trends beforehand and it makes the best decision possible with respect to rescheduling, allocation of slack and assignment of modulation levels. From the pool of entities, it picks out the entity which can give the highest energy gain for each additional unit of slack allotted to it.

This happens iteratively till either the available slack finishes or the available entities cannot be scaled down any further. Comparison with Clairvoyant scheme can offer more insight about overheads with respect to wrong decisions made by the reshuffle scheme.

3.2.8 Performance Evaluation

We evaluate the performance of our proposed heuristics in order to quantify their energy saving capabilities. For the purpose of comparison, we use the following two algorithms: baseline algorithm (GSS) proposed in [23] and the Clairvoyant algorithm which would give the optimal results given the complete knowledge of the future. We simulate all the schemes using a custom C simulator. We use energy consumption of the schedule as a performance metric in our simulation studies. The energy values are normalized with respect to the energy consumption of an energy-unaware schedule.

3.2.9 Simulation model and parameters

For duty cycled data collection applications the wireless sensor network is generally organized in the form of a tree. For the purpose of evaluation, we generated a tree with 120 nodes with each node having a random number of children between {0, 5}. The distances between a parent and each of its child are randomly generated between {5, 10} following a uniform distribution. The communication radius of the nodes is taken as 10 m. In order to access the performance of our schemes, we varied the following parameters: percentage of nodes in the network having a redundant message, the size of the observation window and listening frequency assigned to nodes. For each simulation, we use the following parameters :

- *Computation Subsystem:* The CPU is assumed to be the Intel Xscale PXA27x CPU which is used on widely available iMote-2 sensor node. Task Cycles = 1000, frequency f can take values from the set f={624MHz, 520MHz, 416MHz, 312MHz, 208MHz, 104MHz} and the associated power consumption is P = {925mW, 747mW, 570mW, 390mW, 279mW, 116mW} [38].

- *Communication Subsystem:* Relevant data packet size = 1024 bits, redundant data packet size = 144 bits, $N_0 = 4* 10^{-13}$ J, BER = 10^{-6}, β_i=75 nJ, β_j= 100 nJ [52], W= $1MHz$. Modulation level b could range from b= {1,2,3,4,5,6,7,8,9,10}

- *Overhead:* All schemes expend listening energy, E_{listen} = 17.3 μ J [53] for each channel

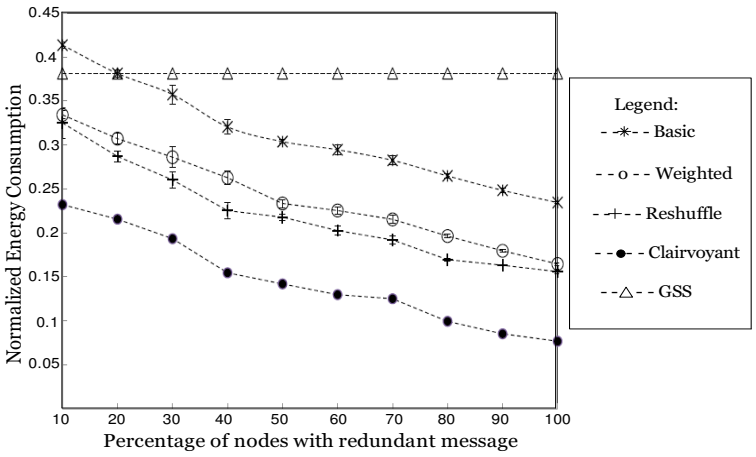

Figure 3.11: Effect of % of nodes having redundant message

sample. Reshuffle schemes compute the schedule once per observation window. The reschedule computation takes $Task_{Reshuffle}$ = 1000 cycles and the schedule broadcast packet size = 168 bits.

In order to simulate a correlated network, we used the tool proposed in [54] to generate randomly correlated data. The redundant or relevant character of the node is decided by the resolution parameter R whose values are varied from 1% to 10%.

3.2.10 Simulation Results

Varying percentage of nodes with a redundant message: Fig. 3.11 shows the performance of the schemes with varying percentage of nodes with redundant message. Energy consumption reduces as the percentage of such nodes increases. Static slack is given as an input to the algorithms. As the static slack increases, nodes get more opportunities to scale down the modulation levels of their messages. We define slack factor (sf) as the ratio of static slack and deadline. Fig. 3.11 shows the performance of the schemes at sf =0.1 and shows the normalized

energy consumption values with 95% confidence intervals with an error bound of 5.306%. The other parameters like observation window size and listening frequency are kept constant at 5 and 1 respectively. All the three schemes give better energy savings as compared to GSS. In cases where the number of redundant nodes are less, Basic scheme leads to worse performance than GSS owing to the channel listening cost. Reshuffle algorithm preforms slightly better

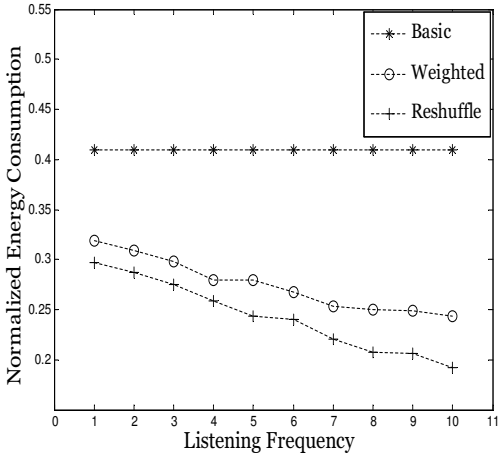

Figure 3.12: Effect of listening frequency on energy consumption

than the Weighted Sharing scheme owing to the cost of schedule broadcast at the end of every observation window. The Reshuffle scheme incurs as small as 33% and as large as 103% more energy than the optimal Clairvoyant Scheme. With careful choices of the observation window size, reshuffle algorithm is shown to be capable of saving up to 40% more energy than the existing GSS scheme [23]. Energy saving directly translates into lifetime extension of the sensors. According to the study conducted in [55], battery life of a Mica-2 mote sending 4 messages per second, at 100% duty cycle, using two AA batteries is 180 hours($\sim$ 1 week). In general, duty cycle of the sensors is kept at less than 4%, and if messages are sent at the rate of one message per second, the lifetime can approximately be calculated as 100 weeks. Given that our proposed reshuffle scheme can save up to 40% more energy than the existing state of the art scheme, the lifetime can be extended to approximately 140 weeks. Please note that this calculation between energy savings and lifetime extension is approximate as it ignores the energy overheads associated with protocol processing and sleep-wakeups.

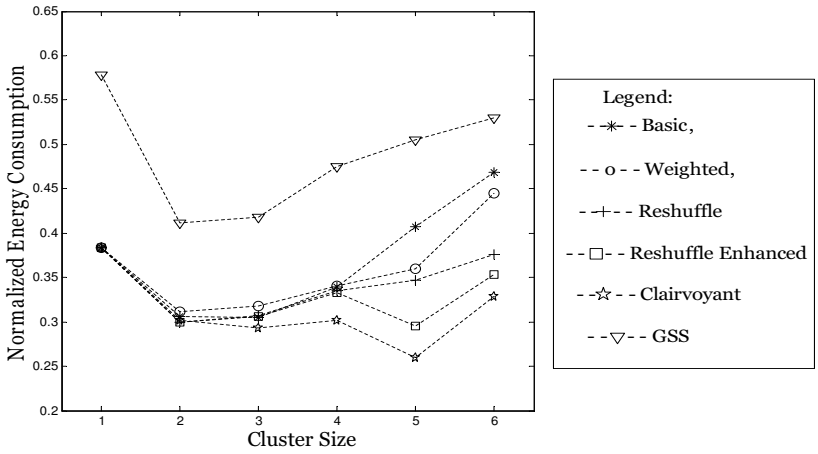

Figure 3.13: Effect of cluster size

Varying the listening frequency of a node: Fig. 3.12 shows the effect of varying the listening frequency on the energy consumption of the network. In basic scheme, since the amount of slack can be pre-calculated, there is no need for *strobed listening*. But in weighted and reshuffle scheme, the frequency of channel sampling affects the energy saving capability. If nodes sample channel more frequently, they will expend more energy in listening but they will also have more opportunities for identifying available slack and utilizing it. We varied the listening frequencies allocated to the nodes in order to evaluate the tradeoff between the two. A frequency value of n means that node listens to the channel n times between the $Start_{Listen}$ time and the scheduled start time, with the channel listening starting at $Start_{Listen}$. The performance is as shown in the Fig. 3.12. The evaluation was done on a network that has 50% nodes having redundant message and slack factor at 0.05. In case of basic scheme, the listening frequency is kept constant at 1. In weighted and reshuffle schemes as the listening frequency goes higher the energy savings due to increased slack availability becomes dominant as compared to the energy wasted in listening. This is because nodes now have more opportunities to identify available slack. Thus, it is evident that the energy savings achieved due to utilization of slack surpasses the energy overhead encountered in identification of available slack.

Varying Cluster Size: Fig. 3.13 shows the effect of varying the cluster size. The sliding window is chosen as 10 and 50% of nodes have redundant message. When each cluster is made

up of only one parent node and one child node, then all the schemes behave exactly alike because there is no peer node to reclaim the slack of a redundant node. The energy saving is due to the suppression of redundant messages only. Also since there is only one child per node, amount of energy expended in overheads is zero. However, this is a very uncommon case because WSNs are rarely deployed in this manner. As the cluster size increases, there are increased opportunities to reclaim slack and also increased overheads.

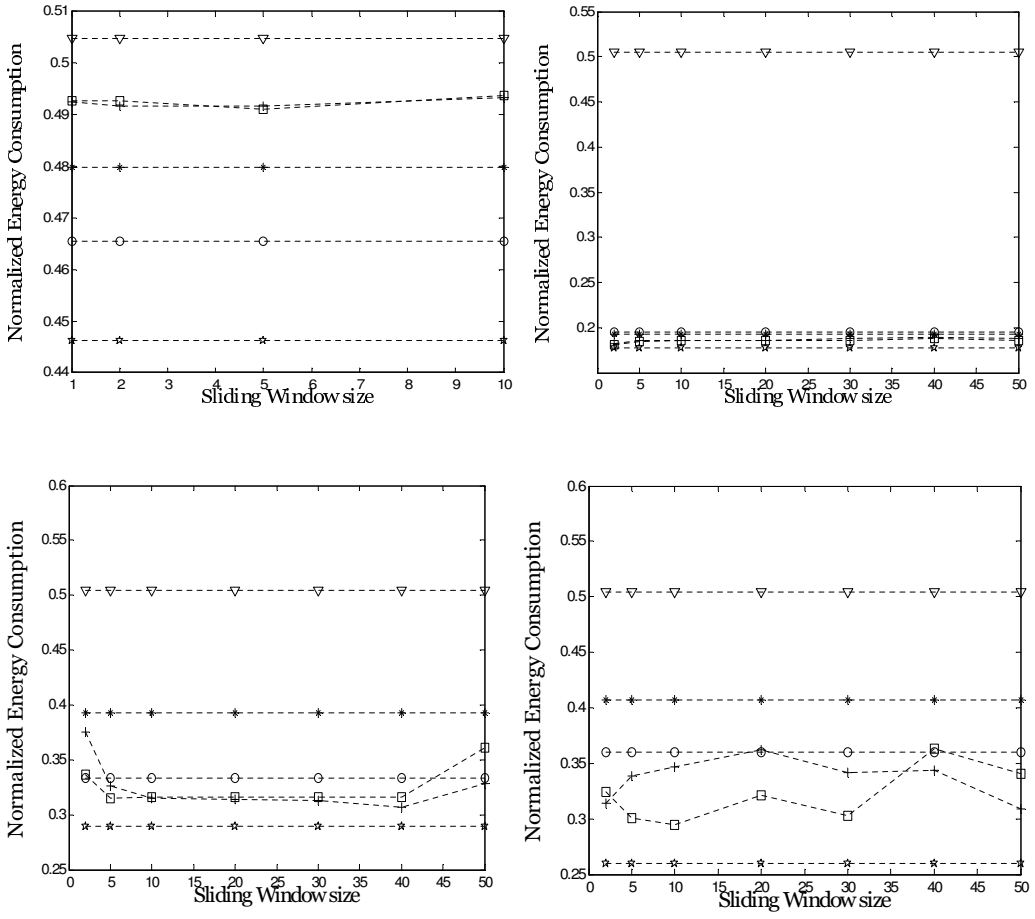

Figure 3.14: Effect of observation window size for Data Pattern 1, 2, 3 and 4

In cases where the cluster size is large and the nodes with redundant message are at the end of the schedule, Basic and Weighted are unable to make use of the slack. Reshuffle and Reshuffle enhanced schemes achieve better energy savings due to relocation of slack to be

utilized by nodes with relevant messages.

Varying Observation Window: In order to assess the effect of knowledge of repeating data patterns, performance of the proposed schemes with respect to varying size of the observation window is as shown in Fig. 3.14. The simulations were performed for 1000 instances in a network having 50% nodes having redundant message, with the listening frequency kept constant at 1 and slack factor at 0.1. For testing performance of schemes in the presence of data traces with repeating patterns, we used data sets similar to [56] where the data fluctuates cyclically. The data sets are processed into square or rectangular pulses to show the redundancy trends. Fig. 3.15 shows the data patterns (DPs) we use for our simulations. The DPs show the varying redundancy-relevancy trend and not the actual values of sampled data. DP1 shows absence of any pattern in any node, with data values varying at every time instant. DP2 shows all the nodes following the same pattern. DP3 shows that 50% of the nodes follow a redundancy trend and the other 50% have no repeating pattern. DP4 shows nodes observing different types of repeating patterns.

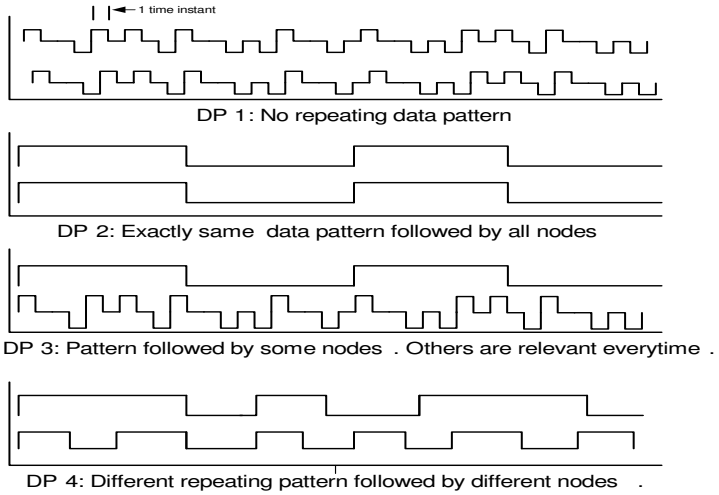

Figure 3.15: Data Patterns

Case 1: No Pattern

Sampling data pattern is shown in plot 1 of Fig. 3.15. Since the data does not follow any pattern, the existence of redundant values is totally arbitrary. In such cases, a prediction based scheme can be of no help. As shown in Fig. 3.14, Weighted behaves best among all the

schemes. Reshuffle and Reshuffle Enhanced perform worse than Basic and Weighted owing to overheads that surpass any random savings. Thus for applications where sensors record consecutive uncorrelated activity without any repeating patterns, Weighted is the best scheme for energy minimization.

Case 2: Same Pattern

Sampling data pattern shown in plot 2 of Fig. 3.15 shows all sensors showing exactly identical repeating pattern. In such cases, all the schemes behave very similar as shown in Fig. 3.14b). Since majority nodes are redundant majority of the time, the basic and weighted schemes attain savings which is the same as what reshuffle and reshuffle enhanced attain with overheads.

Case 3: Mixed Pattern, Uncommon Cases

Plot 3 in Fig 3.15 shows a scenario where a portion of the nodes follow a repeating data pattern and the rest of the nodes exhibit relevant values all the time thus showing absence of any repeating pattern. Given that these are nodes in a cluster, this constitutes an uncommon situation and can occur in cases when some nodes are faulty and hence show irregular behavior. For small observation window sizes, Reshuffle Enhanced performs best in the presence of overheads. But for larger window sizes, reshuffle perform best owing to the fact that it averages the redundancy count over a large window size. Since some nodes do follow a pattern, over large window sizes, these average approximations are much better than reshuffle enhanced or weighted.

Case 4: Mixed Pattern, Common Cases

Nodes follow different repeating patterns. This pattern portrays common scenarios encountered by WSNs for data collection applications. Nodes can be following varying data patterns owing to different locations, differing characteristics of the attribute that they are sampling etc. We see that in such cases, Reshuffle Enhanced is the best scheme. Except when the observation window gets too big as compared to the repeating pattern where reshuffle performs better than reshuffle enhanced.

Effect of Wireless Interference: In this section, we assess the impact wireless interference can have on our proposed schemes. We emphasize that even if slack reclamation is not

feasible at all due to intense interference, energy could be still be saved by avoiding to transmit repetitious sensor readings.

Impact of interference on slack determination: For slack determination, a node i listens to the channel to determine if its peer predecessor node j is redundant or relevant. Notice that i does not need to correctly receive the message and hence interference from simultaneous transmissions do not impact the decision at i. If the signal strength is lower than a predetermined threshold, it is enough for i to decide that the predecessor node j is not actively transmitting and the slack is available for i.

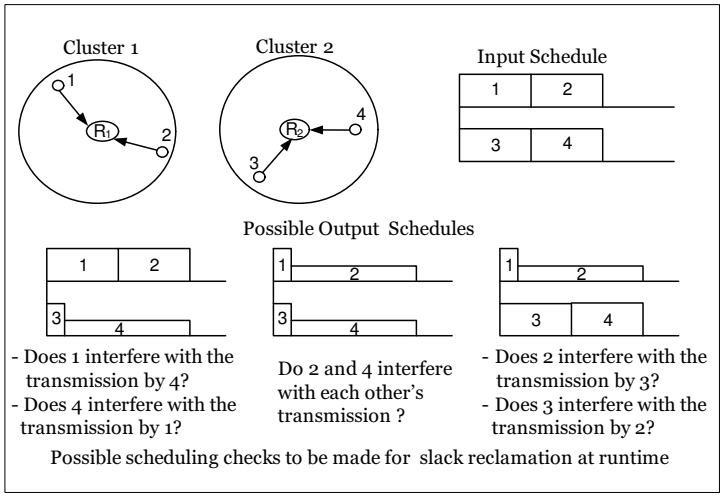

Figure 3.16: Slack reclamation scenarios

Impact of interference on slack reclamation on each scheme: Fig. 3.16 shows an example of an input schedule and cases that arise when this schedule is modified due to runtime slack reclamation.

Basic: Given that all the available slack is reclaimed by only one successor node (Fig. 3.7), the resulting modulation level and hence transmitter power can reduce significantly. The aggregate interference can now impact successful transmissions more than before, negatively impacting performance. This effect will be more pronounced when number of nodes with redundant message are less. It can be avoided by conservatively predetermining the least modulation level that can be attained while still meeting the threshold for successful receptions.

Weighted: Available slack is shared amongst successor nodes (Fig. 3.9), hence signal strength

does not reduce drastically, thus avoiding the performance degradation similar to Basic.

Reshuffle and Reshuffle Enhanced: Both reshuffle schemes behave similarly to Weighted scheme, with the modification that both the reshuffle schemes periodically reschedule the nodes in each cluster, such that nodes with high number of redundant messages are placed at the beginning of the schedule. The rescheduling is done locally by each cluster head and interference can be avoided given that condition in Eq. 3.2 is followed by the topology. However, physical interference modeling would restrict the reshuffling based entirely on the *mostly redundant node first* metric. Under worst case scenario, they would behave either similar or slightly poorer than Weighted scheme.

3.3 CSMA systems: Channel and load state aware CSMA/CA

3.3.1 Introduction

IEEE 802.15.4 has been widely adopted as a standard for the Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs) including WSNs. Carrier sense multiple access with collision avoidance (CSMA/CA) mechanisms have been used for contention based medium access for WSNs because of their energy saving features. However, the absence of adaptation to varying channel conditions leads to energy wastage due to retransmissions by nodes experiencing deep fading. The readings of a sensor node that is deployed for the purpose of data monitoring and control must reach the receiver by the intended deadline for real-time decision making. Each sensor message is either transmitted to its destination before its deadline or is discarded. Higher transmission rates lead to low transmission delays but also incur higher energy. Thus energy consumption and delay have an inherent tradeoff.

Dynamic Modulation Scaling (DMS) has been studied as an effective rate adaptation scheme for real-time networks [[25]],[[23]]. It trades off transmission energy with transmission latency in networks employing wireless communications. We propose an adaptation of the slotted CSMA/CA described in IEEE 802.15.4 combined with the Dynamic Modulation Scaling

scheme for a soft real-time WSN. Depending on channel state and the load index at a node, the best modulation level can be selected such that the energy consumption can be reduced and the goodput of the system can be increased. Our adaptive CSMA/CA is implemented in an entirely distributed manner. The runtime variations occur as a result of integrating the channel and load state information at each node. We focus on the combined reduction in energy consumption and transmission delay while considering the time varying channel and the real-time nature of the messages. However considering that the two metrics have an inherent tradeoff, we propose an energy-delay metric which is a weighted linear combination of the transmission energy and transmission delay. At times when the node is heavily loaded, the metric selects the low delay option for faster data transmission. Alternatively, when the node is lightly loaded, the metric selects low energy option to reduce energy consumption.

3.3.2 Related Work

The effect of fading in wireless channels is known to reduce goodput and increase the amount of power wasted in retransmissions [57]. Channel-aware variable rate systems [18] have been proposed to enhance throughput in wireless environments. But the focus has been on reduced latency and increased throughput rather than energy consumption.

In CSMA/CA systems, several existing protocols use the contention window adaptation method [16] and [17]. But in both the cases, the contention window optimization needs to be done by a centralized coordinator incurring overheads. For the CSMA/CA systems, the absence of channel adaptation and load adaptation results in energy wastage due to retransmissions and low success rate. We propose an Adaptive-CSMA/CA protocol which accounts for varying channel and load conditions at a node by influencing the selection of either low energy or low delay transmission option.

In [58], a channel varying scheme is devised for a TDMA system, where messages are postponed till the channel transitions into a good state. However, a waiting mechanism like this is not suitable for real-time messages.

In order to improve the energy and delay efficiency of a network using contention based

schemes, several protocols use the contention window adaptation method. In [16], authors present separate optimal contention window sizes for energy optimized and delay optimized scenarios. In [17], authors use the combination of queue and channel state information. But in both the cases, the contention window optimization needs to be done by a centralized coordinator incurring overheads.

DMS has been used for the energy efficient real-time scheduling of messages in [23] and [46] etc. We propose the use of DMS for energy and delay efficient operation in the existing IEEE 802.15.4 protocol. Thus, our work differs from the existing research as we consider the combined effects of channel variation with the instantaneous load for the joint reduction of energy consumption and delay of a soft real-time WSN in a distributed manner.

3.3.3 System Model

3.3.3.1 PHY Layer

We use the frequency flat Rayleigh fading channel to model the time varying fading wireless channel between two devices. Each link is assumed to be constant during a packet transmission and vary independently according to a Rayleigh distribution. In order to make the communication system energy aware, we use DMS [[25]] which uses Quadrature Amplitude Modulation (QAM) as the transmission scheme. Let k denote the number of bits per modulation constellation symbol. The basic idea of DMS is to vary the number of bits per symbol while keeping the symbol rate constant. For a packet of fixed size, this results in an energy-delay pair for each modulation level. At higher modulation level, a packet can be transmitted at higher energy than a message at lower modulation level. However, the transmission delay of the message at higher modulation level is less than that of a lower modulated message. The best modulation level can be chosen depending on whether the packet needs to be transmitted in an energy-efficient manner or a delay-efficient manner. As we explain later, this tradeoff can be leveraged by noticing that at times operating at higher modulation level is beneficial and at other times the reverse could be more efficient. Table 4.1 specifies relevant symbols used in the chapter.

Table 3.1: Relevant Symbols

k	Modulation Level
d	Transmitter-Receiver distance
G_R	Antenna gain
λ	Wavelength of the transmitted signal
A_R	Effective area of the antenna
SNR	Signal to noise ratio
R_s	Symbol rate
h	Channel gain
W	Channel bandwidth(Hz)
P_b	Bit error probability
E_{CCA}	Energy consumption per channel sample
E_{ACK}	Energy consumption per acknowledgement

In order to reliably transmit a packet from source to destination, the transmitted signal power must be enough to combat the erroneous channel. We fix the bit error probability, P_b, for reliable transmission as 10^{-6}. Given the instantaneous channel gain, h and the Signal to Noise ratio (SNR), the required bit error probability in case of QAM can be calculated as [36],

$$P_b \geq \frac{4}{k} \cdot Q\left(\frac{3 \cdot h^2 \cdot SNR}{(2^k - 1) \cdot R_s}\right)^{\frac{1}{2}} \tag{3.11}$$

where Q function is defined as, $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2}\, dy$
and k can take values from the set $K = \{2, 4, 6, 8, 10\}$.

Alternatively, by fixing P_b, we can find the minimum received SNR required for the signal to be correctly demodulated at the receiver for each value of k as,

$$SNR_{min} = Q^{-1}\left(\frac{k \cdot P_b}{4}\right) \cdot \frac{2^k - 1 \cdot R_s}{3 \cdot h^2} \tag{3.12}$$

From the received SNR, the transmitted power, P_{tx}, can be found as

$$P_{tx} \geq \frac{SNR_{min} \cdot A_d \cdot N_0}{A_R} \tag{3.13}$$

$$A_R = \frac{G_R \cdot \lambda^2}{4 \cdot \pi} \tag{3.14}$$

where the effective area in which power radiates is given by

$$A_d = c \cdot d^\alpha \tag{3.15}$$

where c is a constant and α takes values between 2 to 5. The transmitted power values P_{tx} will be calculated for each k. Given that each transmitter has a limit on the maximum output power value, P_{max}, the values of k for which $P_{tx} \geq P_{max}$ will be discarded. The corresponding channel transmission rate in bits per second can be calculated as Wk. The time, t, taken for transmitting an L-bit message in the wireless medium is calculated as $\frac{L}{Wk}$. The energy per transmitted signal can be calculated as $P_{tx}.t$.

Thus the information about the current state of the channel decides the appropriate power level for the transmitted signal and the possible modulation levels.

3.3.3.2 MAC Layer

The IEEE 802.15.4 MAC standard specifies a beacon enabled mode for the purpose of channel state estimation. We consider the active part of the superframe to be made up of beacon slot and Contention Access Period (CAP) slots only. For superframe structure, we set beacon order, BO = 5 and superframe order, SO = 0, which gives a duty cycle of 3.25% and the duration of the active part of the superframe as 15.36 ms. All other CSMA/CA parameters are assumed to be the default values as defined in IEEE 802.15.4 standard.

IEEE 802.15.4 does not have any option for finding out the channel state, except from the beacon transmission at the start of every superframe. The nodes in the cluster receiving the beacon can calculate the Link Quality Indicator which can be used to find the channel quality [57], [59]. The channel under consideration is a time variant channel. Considering that the frequency of beacon transmission depends on the values of BO and SO which are defined by the application, the beacon might not provide up to date information about the channel quality. For such cases, we propose the inclusion of multiple beacon transmissions per superframe depending on the coherence time of the channel.

Coherence time is a statistical measure of the time duration over which the channel impulse response is essentially invariant [36]. The coherence time is inversely proportional to the maximum doppler shift. To illustrate the wide variation in coherence time, consider two sample application scenarios: a sensor network deployed for the data monitoring of an electrical

transmission line [4] and one that is deployed for wildlife monitoring. In the transmission line application, there could be vehicles moving at high speed (≈45km/hr) in the vicinity of the transmission towers and hence the maximum doppler shift will be greater (typically of the order of 100 Hz). In the case of wildlife monitoring, if we assume that the entities are moving at a velocity of (≈4.5 km/hr), then the maximum doppler shift will be of the order of 10 Hz. The coherence time can be calculated as 4.23 ms in the former case and 42.3 ms in the latter case. Given the different time scales with which the channel might vary, a single beacon transmission per superframe may lead to obsolete channel state information in the former case. Therefore, in our scheme, we include multiple beacons in the active period of the superframe depending on the application and the coherence time of the channel. The inclusion of extra beacon frames leads to the added overhead in terms of extra energy consumption for the transmission and reception and also adds to the delay. However, as shown by our simulations, this overhead is shadowed by the energy saved due to channel aware transmissions.

3.3.4 Channel and load state aware CSMA/CA

With the advent of new sensor technologies, it is possible to integrate a large number of sensors in one sensor node. We assume here that each node is equipped with a suite of sensors, maximum being m. Given the application requirement, the sampling frequencies could be predecided and may be different from one another. For example, while monitoring the electrical overhead transmission lines, temperature variations on the transmission line happen at a much smaller time scale than the variation in tilt of the transmission tower. Also sensors sample the environmental parameters which tend to change gradually. For slow varying phenomenon, the successive sensor readings might not differ from each other. In such scenarios, the repeating sensor measurements can be discarded to avoid expenditure of communication energy. Thus after each sampling the reading is compared with the previous reading. A close match results in discarding the redundant message.

Each sensor message, i in the suite is characterized by the tuple $\{G_i, D_i, P_i\}$, where G_i is the generation time of the sensor message, D_i is the deadline of the message and P_i is the

period of the message. All the sensors in the suite sample periodically. At each period, the sensor reading constitutes a message. If relevant, this message needs to be transmitted before the deadline.

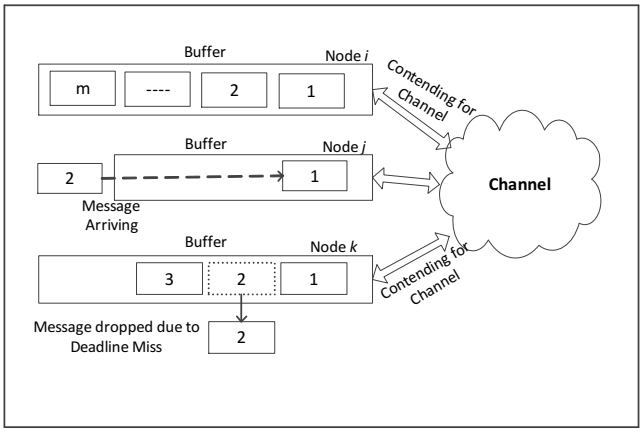

Figure 3.17: Channel Contention amongst nodes

In contention based access protocols, the nodes start contending for a channel in the active period of the superframe, if they have at least one backlogged message. During the contention period, the message stays in the node buffer till either the node wins contention and transmits the message. Else if the message is not transmitted by its deadline, the message is dropped. Thus at any time in the sensor buffer there can be at most one message from each sensor stream and in total there can be a maximum of m messages. The different scenarios are represented in Fig. 3.17. During the contention period, the node might get more messages from other sensors in its suite. Also varying deadlines means different sensors contribute differently to the load with the contribution being inversely proportional to the period. Load index of a node is calculated as,

$$LI = \sum_{i=1}^{m} \frac{WCTx_i}{P_i} \tag{3.16}$$

where $WCTx_i$ is the worst-case transmission time of a message. The load index of a node is the sum of the utilization demands of the messages in its buffer. After each message transmission, the load index is reduced. In order to maximize this reduction, the message with the smallest P_i must be picked for transmission.

Consider a network of nodes contending for exclusive access to the channel. The method of contention is the same as the slotted CSMA/CA described in IEEE 802.15.4 specifications [60]. Fig. 3.18 shows the sequence of steps followed by each node for the joint selection of message and modulation. Each node follows these steps after winning contention and acquiring the channel. Once a node wins contention, there are two choices to be made:

- which message should be picked for transmission, and
- what modulation level should be used to transmit this selected message.

With respect to the message selection, the node makes a greedy decision of selecting the message with the least P_i, so that its load index can be reduced by the maximum amount (step 2). The channel state influences the required transmission power and hence the transmission energy to meet bit error restriction (step 3). Given the power restrictions, it means selecting which modulation levels are possible (step 4). Amongst the possible modulation levels, the load index influences the selection of the best (step 8).

In order to select the best modulation level for transmission, we propose a novel metric called the *energy-delay metric*, M_i as,

$$M_i = \beta \cdot E_i + \gamma \cdot D_i \tag{3.17}$$

where E_i is the normalized transmission energy consumption and D_i is the normalized transmission delay, at modulation level i.

$$\gamma = LI, LI \leq 1 \tag{3.18}$$

$$\beta = 1 - \gamma \tag{3.19}$$

The modulation level i for which the metric M_i has the minimum value is the best modulation level under the current channel and load conditions. Since the objective is the combined reduction of energy and delay, the minimum value of the metric selects the best option. If the node is heavily loaded, the modulation with the least transmission time (D_i) is selected by

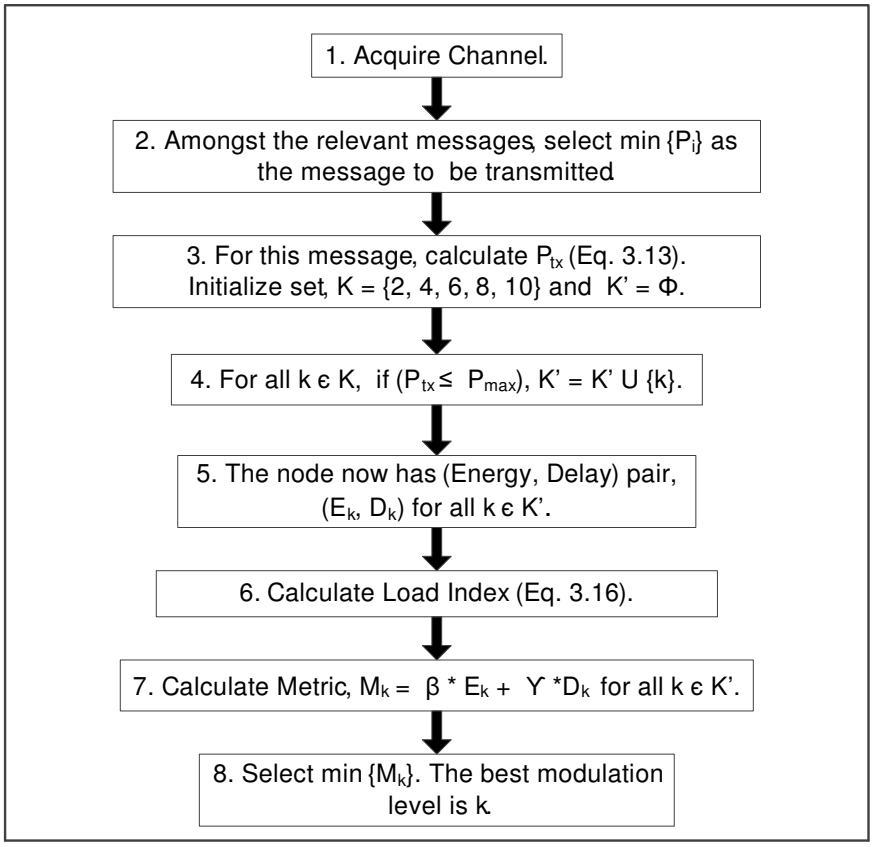

Figure 3.18: Joint selection of message and modulation at a node

the metric. Alternatively, if node is lightly loaded, the modulation with the least transmission energy (E_i) is selected. Thus, channel and load variations influence the selection of either low energy or low delay transmission.

3.3.5 Performance Evaluation

Due to the unavailability of devices that implement DMS, we build a custom Java simulator for performance evaluations. We perform extensive simulations to quantitatively assess the performance of our scheme, Adaptive-CSMA/CA and the one used in IEEE 802.15.4, referred to as CSMA/CA. For fair comparison, the transmission energy and transmission time of each message in CSMA/CA is calculated with respect to the QAM scheme with $k = 2$. The performance metrics of interest are normalized total energy consumption and success ratio. Total energy consumption, E_t is computed as,

$$E_t = (P_{tx} + n \cdot P_{rx}) t_{Beacon} \cdot i + 2 \cdot E_{CCA} + (P_{tx} + P_{rx}) t_{Data} + E_{ACK} + C \quad (3.20)$$

where n is the number of nodes in the system and i (=3) is the number of beacon transmissions per superframe for up-to-date channel state information, t_{Beacon} and t_{data} are the beacon and data transmission times respectively, P_{tx} and P_{rx} are the transmission and reception power per message respectively and C is the energy overhead for transmission and reception circuitry.

Success ratio is the ratio of successful transmissions to the total number of messages generated. Generally, unsuccessful transmissions could be either due to the transmission of message while the channel is in a deep fade, collision with another packet or a deadline miss while waiting in the node buffer. Considering that the wireless link is assumed to be constant during a packet transmission and the packet transmission power is adapted to be enough to combat the channel according to Eq. 3; and collisions are avoided by the CSMA/CA protocol, the loss of a packet is only due to violation of its deadline due to the delay in its transmission. Lower the transmission delay of a message, the faster the node will relinquish the channel and the earlier it is available for another contending node. Thus, success ratio affects the transmission delay.

3.3.5.1 Simulation Parameters

We consider a star topology with a Personal Area Network (PAN) coordinator and 10 nodes. This is because the other topology proposed by the standard, peer-to-peer topology,

suffers from beacon frame collisions when the beacon enabled mode is used [61]. The nodes are at a distance of 50m from the PAN coordinator. The nodes being equidistant, removes the effect of path loss so that the effect of fading can be studied. The values of other parameters are as follows: R_s= 3Mbps, N_0 = 4.11* 10^{-16}, G_R = 3.3dB, λ = .125m, $P_b = 10^{-6}$, data packet size, L = 1024B, beacon packet size, B = 200B, $E_{CCA} = 17.3\mu$J [53], $t_{Beacon} = 266\mu$ sec, P_{tx} and P_{rx} for beacon = 80.5mW, WCTx = 1365 μsec. P_{tx} and P_{rx} for data packets are calculated depending on the channel gain. Each point on every plot is an average of 20 simulation runs. For a plot, the energy values are normalized with respect to the highest energy

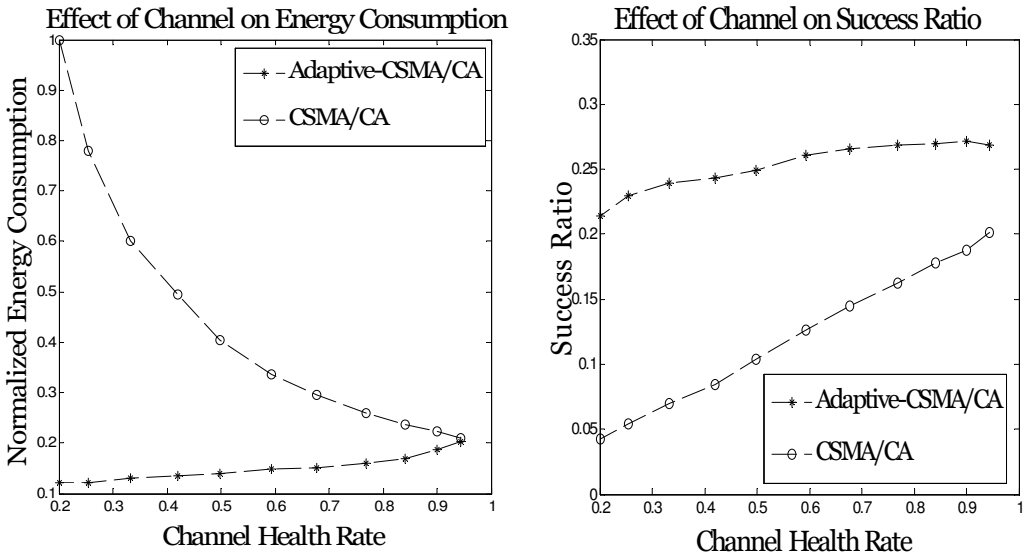

Figure 3.19: Effect of channel variation

consumed amongst the two schemes in that plot. We simulate diverse scenarios and our results archive better energy savings and success ratios than the existing CSMA/CA scheme.

3.3.6 Effect of Channel

Fig. 3.19 compares the performance of Adaptive-CSMA/CA over CSMA/CA with respect to energy consumption and success ratio. We use the threshold model used in [57] to gauge the channel quality. Channel health rate is defined as the ratio of the number of times channel was in good condition to the total number of times the channel was sampled for transmission. For each node load is kept constant at 0.25 and percentage of redundant nodes is constant at 0%.

Since CSMA/CA never adapts to the varying channel, huge amount of energy is wasted during retransmissions. Because of the added delay due to retransmissions, the messages waiting in the buffer miss their deadlines, thus reducing the success ratio.

3.3.6.1 Effect of Number of Contenders

The number of contenders effect the contention delay experienced by nodes. With increasing contention delay, the number of messages missing their deadlines increase which leads to

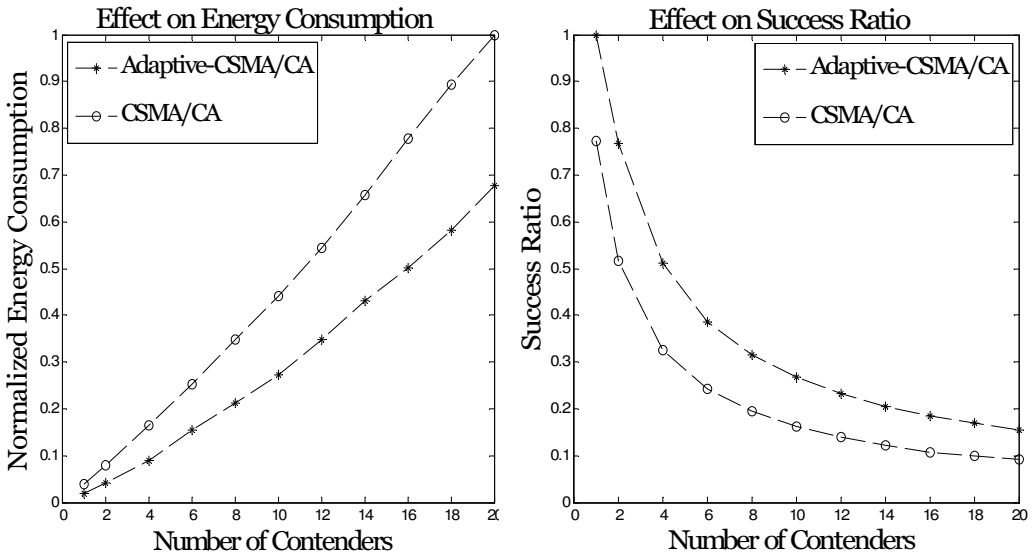

Figure 3.20: Effect of variation in number of contenders

lower success ratios. Also, as contention delays increase, more messages are waiting in the buffer, which increases the load indices of nodes. Greater load indices lead to nodes selecting higher modulation levels, hence increasing the energy consumption. In both cases, as shown in Fig. 3.20, Adaptive-CSMA/CA performs better than CSMA/CA.

3.3.6.2 Effect of Load

Fig. 3.21 shows the effect of load variation, where the performance is characterized with delay-only and energy-only schemes too. For these two cases, the modulation level chosen is always the one with the least transmission delay and the least transmission energy respectively. The performance of these two schemes shows the influence of ignoring the effect of load index, while the performance of CSMA/CA is due to the use of a single fixed modulation scheme. The results reinforce the finding that metric based Adaptive-CSMA/CA strikes a balance between low energy and low delay schemes. However, in cases of high load, Adaptive-CSMA/CA trades off energy consumption for improvement in success ratio. At high load conditions, the metric decides for higher modulation levels, thus improving the success ratio but expending more energy,

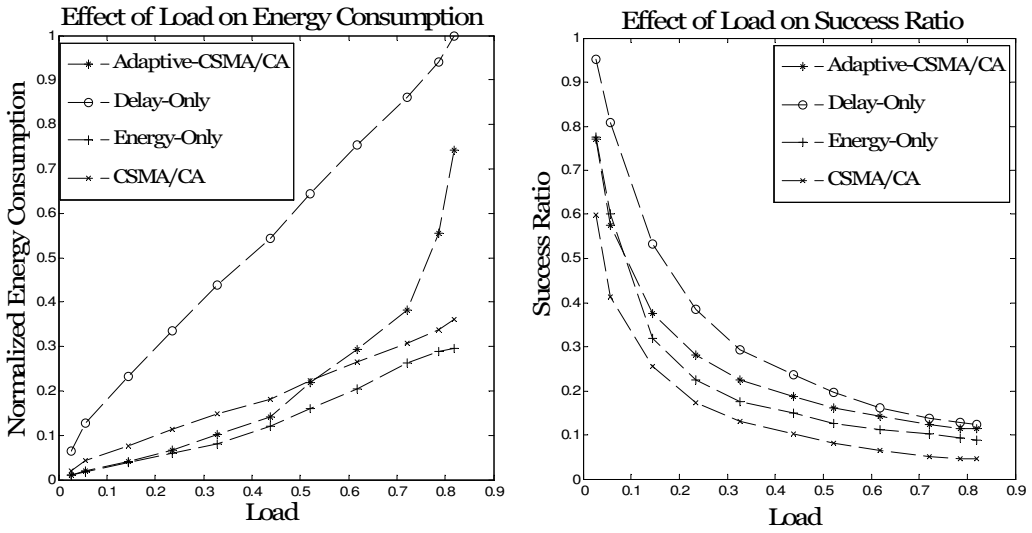

Figure 3.21: Effect of variation in load

3.3.6.3 Effect of Redundancy

Fig. 3.22 shows the effect of varying number of redundant nodes. As expected, energy consumption reduces linearly as the number of relevant messages decrease. The success rate shows similar increasing trend. In both cases, Adaptive-CSMA/CA performs better than CSMA/CA.

3.4 Conclusions

In this chapter, we addressed resource management in nodes of a WSN by reclaiming unused resources at runtime. We considered two MAC schemes namely, TDMA and CSMA/CA.

With respect to TDMA based systems, we showed that temporal correlation in sensor data can be exploited in a lightweight manner to generate runtime slack which can then be traded off to reduce energy consumption. We evaluated the performance of our schemes with respect to several factors that can affect the energy consumption like percentage of nodes having redundant message, listening frequency and observation window size. Our simulation results show that the proposed schemes provide much better energy savings as compared to the baseline algorithm [23]. With careful choices of the observation window size, reshuffle

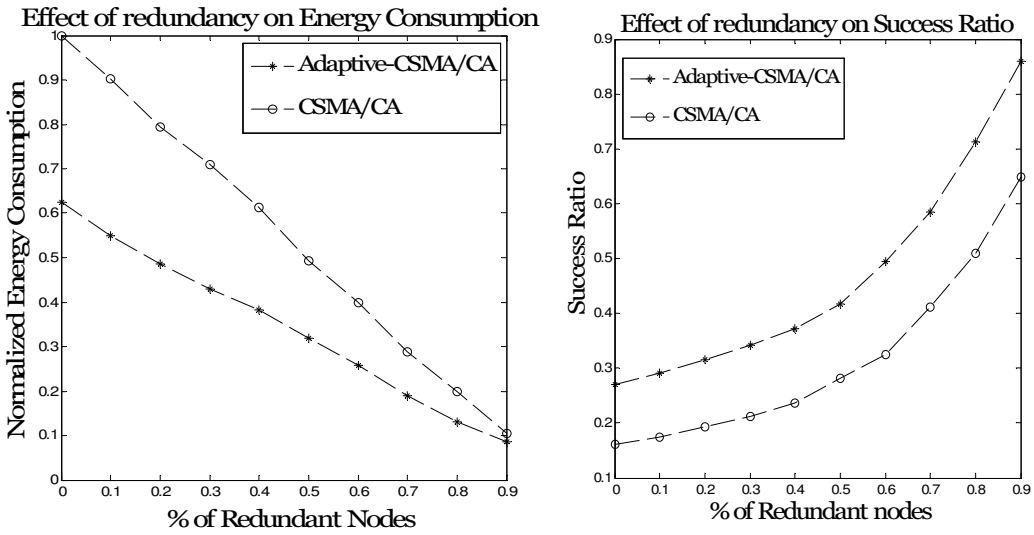

Figure 3.22: Effect of variation in sensing redundancy

algorithm is shown to be capable of saving up to 40% more energy, despite the extra overhead incurred. Even the very simple basic scheme can have better performance than the baseline scheme.

With respect to CSMA/CA based systems, we showed that adapting the transmission strategy with respect to the instantaneous load and channel by dynamically changing the transmission energy and rate through the use of DMS can lead to lower energy consumption and higher success ratios. Our simulation results show that the proposed Adaptive-CSMA/CA scheme results in significant improvement in energy consumption under varying channel conditions and higher success ratio under varying load as compared to the CSMA/CA specified in IEEE 802.15.4.

CHAPTER 4 Wireless Network Design for Transmission Line Monitoring in Smart Grid

4.1 Summary

In this chapter, we develop a real-time situational awareness framework for the electrical transmission power grid using WSN. The low power, low data rate devices cause bandwidth and latency bottlenecks. Our objective is to design a wireless network capable of real-time delivery of physical measurements for ideal preventive or corrective control action. We observe that existing research relies heavily on the assumption of a symmetrical underlying system. Instead, we propose a generic formulation that addresses several real-world concerns and provide ways to extend the formulation to address future deployments. We study a hybrid hierarchical network architecture composed of a combination of wired, wireless and cellular technologies that can guarantee low cost real-time data monitoring. For network design, we formulate an optimization problem with the objective of minimizing the installation and operational costs while satisfying the end-to-end latency and bandwidth constraints of the data flows.

4.2 Background

Currently, there is an impending need to equip the highly vulnerable, age old electric power transmission line infrastructure with a high-performance data communication network that supports future operational requirements like real-time monitoring and control necessary for smart grid integration [62], [63]. Wireless sensor based monitoring of transmission lines provides a solution for several of these concerns like real-time structural awareness, faster fault localization, accurate fault diagnosis by identification and differentiation of electrical

faults from the mechanical faults, cost reduction due to condition based maintenance rather than periodic maintenance etc. The use of sensor networks has been proposed for several applications like mechanical state processing [64], [65] and dynamic transmission line rating applications [66].

To monitor the status of the power system in real-time, sensors are put in various components in the power network [67], [68], [69]. These sensors are capable of taking fine-grained measurements of a variety of physical or electrical parameters and generate a lot of information. Network design is a critical aspect of sensor based transmission line monitoring due to the large scale, vast terrain, uncommon topology and critical timing requirements.

The rest of this chapter is organized as follows. Section 4.3 explains the related work followed by sensor network design in section 4.3. Section 4.3 presents evaluation studies and Section 4.3 concludes the chapter.

4.3 Related Work

Managing the communication burden and resulting data latency is essential for efficient analysis and fast control responses and calls for distribution of intelligence throughout the infrastructure [62], [10]. Given the vast geographical expanse of the transmission line infrastructure, wireless networking presents a feasible and cost effective solution for transmission of monitoring data [63]. Several works [19], [3], [4], [20], [21] and [22] propose to improve the state of the art in transmission line monitoring by harnessing the power of wireless sensor networks for real-time monitoring and control. In these works, the goal is to deploy multiple different sensors in critical and vulnerable locations of the transmission line to sense mechanical properties of its various components and transmit the sensed data through a suitable wireless network to the control center. However, most of these works address this theme at a very high level of abstraction. Small scale real world deployments of wireless sensors include tension monitoring using load cells [70], power donut for conductor surface temperature monitoring [71], sagometer [72] etc.

Authors of [4] and [20] were the first to propose a two level model specifically for support-

ing the overhead transmission line monitoring applications. But considering the topological constraints posed by the transmission lines, the low bandwidth, low data rate wireless nodes would fail to transmit huge amount of data in a multihop manner. The hierarchical model proposed in [19], offers a very expensive solution with the idea of deploying cellular transceivers on every tower. While such a network can provide extremely low latency data transmission, this model is highly cost inefficient as it incurs huge installation and subscription costs. The only work that addresses the problem of finding optimal locations of cellular transceivers is presented in [3].

In [3] authors develop a quadratic equation based solution for finding the optimal locations of cellular transceivers aiming to minimize the delay in information delivery. We contrast this work on the following grounds:

- The formulation presented in [3] relies heavily on symmetry. The underlying network infrastructure and the cellular infrastructure is assumed to be symmetric and available at all times. Further, it is assumed that all transmission towers are identical and transmit the same amount of data. However, several factors bring in asymmetry as enumerated below:
  - Sparse cellular coverage (due to unavailability of cellular towers in the area) or cellular outage.
  - Variation in the amount of data transmitted by the towers in lieu of its location or situation.
  - Irregular terrain in certain regions of the transmission line might prevent the usage of any wireless device forcing the use of only cellular network.
- The analysis presented in [3] considers minimizing delay as an objective. While cost consideration is mentioned in the chapter, deployment and maintenance costs are not used as factors restraining the number of cellular transceivers. In the absence of cost constraints, the latency can be minimized by putting a cellular link on each tower which can be highly cost inefficient.

- The method presented in [3] utilizes a quadratic equation to find out the number of cellular enabled towers required and subsequently the location of such towers. Roots of quadratic equation are rounded off to depict the number of cellular enabled towers, which must be an integer. This rounding off can lead to incorrect results. Also, the factors such as latency and bandwidth which affect the placement of cellular transceivers (referred to as representative node in [3]) in a group are not considered simultaneously, rather bandwidth constraints are considered as an afterthought. This leads to suboptimal results.

The task of designing a robust wireless data communication network involves consideration of various factors such as latency, resiliency, security and bandwidth constraints. While low cost wireless sensor nodes enable large scale deployment and minimal maintenance operation, these low data rate wireless links can prove to be a bandwidth bottleneck when considering the topology of the transmission line network. Transmission towers are deployed in a straight line forming a *linear network* [19] spanning hundreds of miles. An intelligent choice of technologies needs to be made such that the required bandwidth is provided for the intended data to reach its destination in a timely manner.

Fig. 4.1 shows our proposed framework. It enumerates an array of challenges and constraints associated with monitoring a wide area network like transmission lines. Necessary control or maintenance decisions can be taken once the sensor measurements are validated and the physical structure is critically assessed for the presence of faults.

The linear network topology proves to be a major challenge for wireless network design with respect to latency constraints and bandwidth constraints. Performance evaluation of the linear network model [73] shows that successful delivery ratio of the packets from the nodes far away from the substation is found to be much less than that of nodes near the substation

because packets from a farther node have to travel a longer distance and the rate of collision is higher. The effective monitoring of a large transmission line network requires a hybrid communication infrastructure.

This hybrid infrastructure can be a combination of wired (copper cable/optical fiber) and

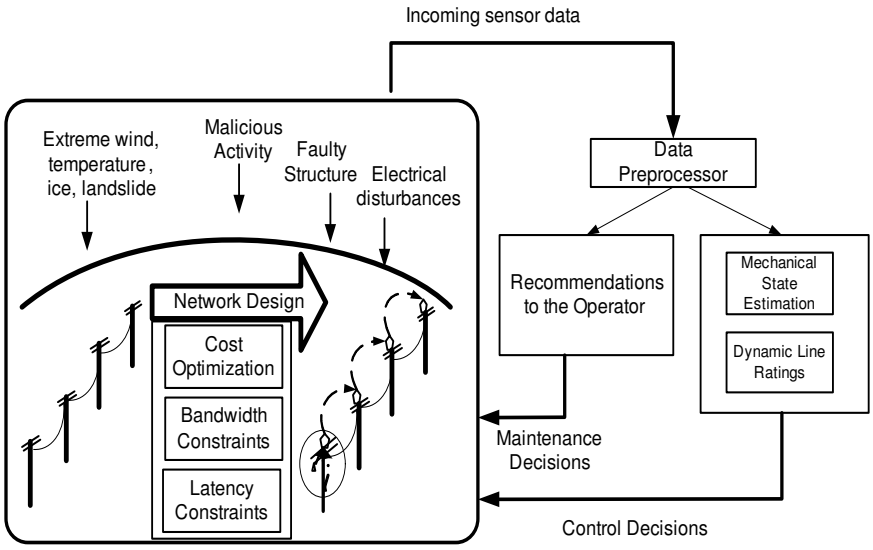

Figure 4.1: Sensor Network Design Framework

wireless (cellular/IEEE 802.15.4) standards to enhance the capability of the overall network to meet newer requirements based on emerging smart grid applications.

In this chapter, we formulate a hybrid hierarchical network design problem that can provide cost effective data transmission while at the same time respecting the bandwidth, delay and connectivity constraints. We formulate a placement problem to optimize the number and location of the cellular enabled towers to significantly reduce the operational and installation costs while respecting all the constraints.

4.4 Three Level Hierarchical Network

We propose a hierarchical three level wireless network model for time critical applications. Each level is equipped with an array of sensors and transceivers with varied capabilities such that together they achieve the required behavior. The design involves the installation of a private WSN of low cost, low data rate links, utilization of the existing SCADA network, and a wide area network such as cellular network comprised of expensive but high data rate links. The proposed network makes use of the existing SCADA links (optical fiber) for communication between substations and control center and strategically utilizes the existing cellular network for data transmission from certain transmission towers directly to the control center. A set of

wireless sensors on each tower is installed as part of the private WSN.

depicts a power transmission corridor with a number of transmission towers, two substations one at each end of the transmission line and a control center. Each level of the network forms a cluster supporting *many to one* communication from all the nodes in the cluster to the cluster head.

The first level of the network is responsible for collecting information about the tower. It is composed of sensor nodes installed in each transmission structure forming a Sensor Array in Tower (SAT). This SAT consists of an array of sensor modules such as tension sensors, accelerometers, temperature sensors, tilt sensors, motion sensors, vision-based sensors, and infrared sensors etc. similar to [3], [4]. Each tower is equipped with a more sophisticated relay node with enhanced computation and communication capabilities. Data from each sensor in the SAT is transmitted to the relay node. The relay node is responsible for compressing the data received from the SAT and transmitting it to the higher level.

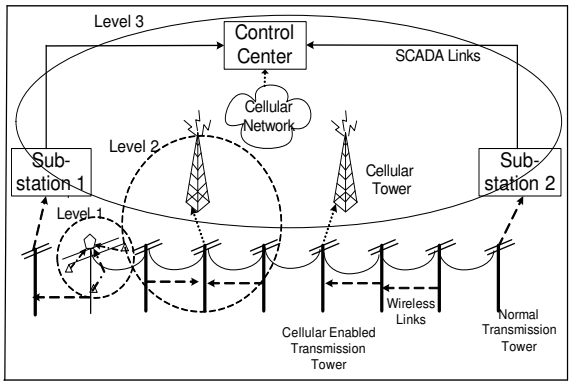

Figure 4.2: Hierarchical Network Structure

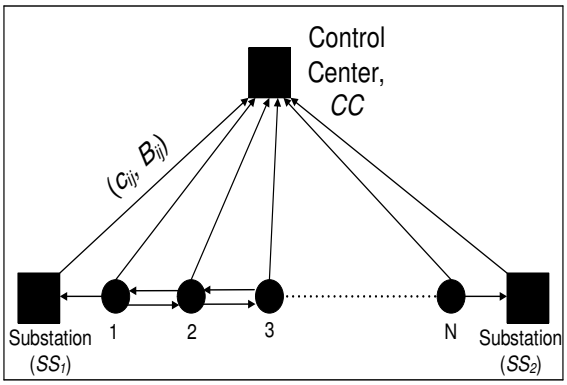

Figure 4.3: Placement Graph

The second level of the network is responsible for transmission of data from towers that are far away from the substations. Consider a segment composed of a few towers in the middle of the transmission line network. Data from these towers cannot reach either of the substations due to limited bandwidth of the intermediate wireless links. In such cases, enabling one of these towers with cellular capability can provide a feasible solution as shown in Fig. 4.2. It is to be noted that it is not required to enable all towers with cellular technology as proposed in [19]. The second level is thus composed of segments of such towers transmitting their

aggregated information to the cellular enabled transmission tower which acts as the head of their segment. The cellular enabled tower is a transmission tower equipped with an additional cellular transceiver along with the relay node. This cellular transceiver offers an alternative way to deliver the tower's data directly to the control center through a high bandwidth, low latency cellular network.

The third level of the hierarchical network is composed of a single cluster consisting of two substations and the cellular towers. The control center acts as the cluster head. Thus, level 1 operates at each tower, level 2 operates at the granularity of a group of towers. The size of the group will be dictated by the wireless link bandwidth and the required end to end latency. Level 3 operates at the level of the whole network where substations and cellular towers transmit to the control center. Table I summarizes the characteristics of various communication standards used in this chapter.

4.4.1 Placement Problem Formulation

In order to provide cost optimized operation in delay constrained and bandwidth constrained linear networks, the strategic placement of cellular transceivers becomes very crucial. While the cellular transceivers provide low latency high bandwidth links, they are costly to install and the subscription charges can dominate the operational cost of the network. On the other hand, the wireless Zigbee devices are relatively inexpensive but provide very low data rates. Thus, there is a tradeoff between cost and delay. In this section, we first explain our network model and state the placement problem. Next, we formulate a mathematical program to find the optimal location of the cellular enabled towers.

4.4.1.1 Network Model

In this chapter, the transmission line is modeled as a directed graph, $G = (V, E)$ as shown in Fig. 4.3. V represents the set of vertices and E represents the set of edges in G. The set of vertices consists of N transmission towers, two substations (SS) at the ends of the transmission line and a control center (CC). Thus, the total number of vertices in the graph is equal to

Table 4.1: Types of technologies and their characteristics

Properties	Optical Fiber	Cellular (3GPP LTE)	Wireless (IEEE 802.15.4)
Type of Link in the Network	Substations to Control Center (SS, CC)	Transmission towers to cellular towers (k, CC)	Between towers (k, l)or tower and substation (k, SS)
Bandwidth	10 Gbps	Uplink 75 Mbps, Downlink 100 Mbps	250 kbps
Delay	$\approx$ 1 μsec	$\approx$ 50 msec	$\approx$ 16 msec
Transmission Range	As long as the fiber	100m- 10 km+	10m - 1.5km
Installation Cost	0 (already exist)	$\approx$ 25x-50x	$\approx$ 1x
Operational Cost	$\approx$ 1x	$\approx$ 5x-20x	$\approx$ 2x
Channel Contention	No	No	Yes
Subscription Fee	No	Yes	No

The numerical values presented for installation cost are relative to that of ZigBee and for operational cost are relative to that of Optical fiber.

N+3. Edge set E in G represents the set of communication links which can be wired (SS, CC), cellular (k, CC) or wireless (k, l), where k, $l \in N$. Each link (i, j) in the graph can be described by a tuple (c_{ij}, B_{ij}), where c_{ij} is the operational cost incurred by the link and B_{ij} is the total bandwidth of the link. Given that each transmission tower needs to transmit its monitoring data to the control center, there are N flows in this network with each one having a common destination, CC and common delay constraint, D. Each flow can be characterized by the tuple: $F = (Source, Destination, b, D)$ where, $Source \in N$ is the source node of the flow, $Destination = CC$ is the destination node of the flow, b is the flow bandwidth requirement, and D is the delay constraint to be satisfied by the flow. l_{ijk} is latency incurred by the kth flow on the link (i, j). The latency incurred is the sum of transmission delay and channel access latency.

4.4.1.2 Placement Problem Statement

Given a directed graph $G = (V, E)$ and a set of N flows, find a feasible path for each flow such that the sum of the cost of all the paths is minimized while respecting the delay and bandwidth constraints of each flow. If the minimum cost path chosen by a tower node k$\in$ N includes the edge (k, CC), then a cellular transceiver should be placed on the tower k.

4.4.2 Placement Problem Formulation

The input to the algorithm is the transmission line consisting of N transmission towers and the end to end latency constraint, D. The formulation uses the binary variables $S_{i,j}$, Y_i and $X_{i,j,k}$. $S_{i,j}$ is 1 if link (i, j) is used by at least one flow, showing that operational costs (c_{ij}) are incurred irrespective of if the link is fully utilized or is underutilized. IC denotes the installation cost of a cellular transceiver on a tower, i. Y_i is 1 only if link (i, CC) is used by any flow which means that a cellular transceiver must be installed on node i. $X_{i,j,k}$ represents the choice made for the link (i, j) by the node k. If node k selects edge (i, j) as one of the link in its path, then $X_{i,j,k}$ is equal to 1 otherwise $X_{i,j,k}$ is equal to 0. All the decision variables are binary variables and hence the formulation is an Integer Linear Program (ILP). We used the ILOG CPLEX 12.2 software [34] to solve the ILP. Table II enumerates the different symbols used in the formulation and their meanings. The placement problem can thus be formulated as,

Minimize:

$$f(S_{i,j}, Y_i) = \tau \sum_{(i,j)\in E} c_{ij} S_{i,j} + \sum_{i=1}^{N} IC \cdot Y_i \tag{4.1}$$

Subject to:

$$\sum_{(i,j)\in E} l_{i,j,k} X_{i,j,k} \leq D \quad \forall k \in N \tag{4.2}$$

$$-\sum_{(i,j)\in E} X_{i,j,i} = -1 \quad \forall i \in N \tag{4.3}$$

$$\sum_{k=1}^{N} \sum_{i=1}^{V\setminus CC} X_{i,CC,k} = N \tag{4.4}$$

$$\sum_{(j,i)\in E} X_{j,i,k} - \sum_{(i,j)\in E} X_{i,j,k} = 0 \quad \forall k, i \in N, i \neq k \tag{4.5}$$

$$X_{i,j,k} - X_{j,CC,k} = 0 \quad \forall j \in SS, \forall k \in N \tag{4.6}$$

$$\sum_{k \in N} b_k X_{i,j,k} \leq B_{i,j} \quad \forall (i,j) \in E \tag{4.7}$$

$$X_{i,CC,k} - Y_i \leq 0 \quad \forall i, k \in N \tag{4.8}$$

$$X_{i,j,k} - S_{i,j} \leq 0 \quad \forall (i,j) \in E, \forall k \in N \tag{4.9}$$

$$X_{i,j,k},\ Y_i,\ S_{i,j} \in \{0,1\} \quad \forall i,j,k \tag{4.10}$$

Table 4.2: Symbols used in Placement Problem Formulation

Symbol	Representation
D	End-to-end deadline
l_{ijk}	Latency incurred by the kth flow on link (i, j)
B_{ij}	Total bandwidth of the link (i, j)
b_k	Flow bandwidth for node k
c_{ij}	Operational cost incurred by link (i, j)
IC	Installation cost per cellular transceiver
$S_{i,j}$	Binary Variable. Is 1 if link(i, j) is used by any flow. Indicates presence of a flow on link (i, j).
Y_i	Binary Variable. Is 1 if tower i is cellular enabled. Indicates presence of a cellular transceiver on tower i.
$X_{i,j,k}$	Binary Variable. Is 1 if node k selects edge (i, j) as a link. Indicates presence of kth flow on link (i, j).

Our objective is to minimize the cost function given in Eq. 4.1. Our cost model consists of two types of costs: installation cost and operation cost. Installation cost is a one-time cost of installing cellular transceivers on selected towers. Operation cost is composed of subscription cost and maintenance cost and is recurring in nature. Wireless and SCADA links are assumed to be owned by the transmission company and hence their operational cost is mainly made up of recurring maintenance costs. Cellular links utilize cellular service provided by a third party.

Hence the operational costs for cellular links are made up of a recurring subscription cost to be paid to the third party and maintenance cost. As shown in Eq. 4.1, the total cost is the sum of operational cost of all the paths used for data transmission over the operational period τ and one time cost for installing cellular transceivers on selected towers. Eq. 4.2 restricts the end-to-end latency of every flow to less than or equal to the maximum permissible end-to-end deadline, D. Our proposed formulation is capable of addressing multiple latency requirements. Firstly, consider cases where multiple latency requirements are imposed throughout the operational period. In such cases, the constant deadline D in Eq. 4.2, can be modified to flow specific deadline, D_k. Secondly, consider cases where multiple latency requirements are imposed for only a fraction of the operation time. In such cases, a proactive planning approach can be adopted and resources can be reserved for use during emergency situations demanding higher data rate and lower latency data transfer. Specifically, contingency flows i.e. flows to deal with traffic contingencies can be created during the planning stage, thereby modifying the total number of flows from N to $N^{'}$. The resources reserved for these contingency flows are utilized during emergency situations.

The latency calculations take into account the transmission latency as well as channel access latency experienced by a flow on each link. In order to address fine grained latency calculations, queuing delay can also be taken into consideration, given that sensor measurements may be buffered at one or multiple nodes before transmission. The queuing delay component can be linearly added to the latency component $l_{i,j,k}$ along with transmission latency and channel access latency. Every flow will utilize a set of wireless links and exactly one cellular or SCADA link of type (i, CC) where $i \in N \cup \{ SS_1, SS_2 \}$. On each link, multiple flows can be multiplexed as dictated by the total link bandwidth. The transmission latency of a flow on each link is calculated considering the presence of other flows multiplexed on the same link.

The constraints in Eqs. 4.3-4.6 explain the flow conservation constraints and ensure that exactly one path is selected for a flow generated at node $k \in N$. Eq. 4.3 restricts each tower to be a source of exactly one flow. Eq. 4.4 depicts that CC serves as destination to exactly N flows. Eq. 4.5 and 4.6 ensure flow conservation at each tower and substation respectively,

between the source and destination. Bandwidth of a link (i, j), is denoted by B_{ij} denoting available link specific bandwidth taking into consideration interference on neighboring links. This bandwidth is assumed to be constant over time. We agree that available bandwidth could change due to varying interference levels. However, the proposed method is an offline planning approach and addressing time varying interference is out of the scope of our work. Eq. 4.7 explains that the total flow on each link must not exceed the available link bandwidth. Eq. 4.8 ensures that whenever any link of type (k, CC) is used by any flow, a cellular transceiver will be placed on the tower k ensuing some installation cost. Eq. 4.9 ensures that cost of link (i, j) is counted at most once for the set of k flows that are multiplexed on this link. The last constraint, Eq. 4.10 ensures that the decision variables are binary variables.

Upon interpretation of the output, the location of cellular enabled towers can be found by the value of the decision variable, Y_i. If it is equal to 1, it means that a cellular transceiver needs to be installed on node i.

The operational cost of links of type (SS, CC) is given a minimal value followed by wireless links (k, l) where k, $l \in N$. The operational cost of cellular links (k, CC) are given highest values modeling the high cellular subscription charges. The installation cost for (SS, CC) is set to zero, thus modeling the already present SCADA links and a constant installation cost can be added to the final output to model the fixed installation cost of SAT on each tower. Installation cost of cellular transceivers is added in the objective function. The edges of type (SS, CC) have the least latency amongst all the edges, followed by the cellular links (k, CC), followed by the Zigbee links (k, l) which have the highest latency.

4.4.3 Link Utilization based costs

In our proposed formulation, we assume a fixed periodic subscription cost being charged for each active cellular link. This model could be generalized to represent subscription costs proportional to the link utilization. The existing formulation can be easily modified to address the variable cost structure. Consider the modified objective function as shown below:

f($S_{i,j}$, Y_i, $X_{i,j,k}$) =

$$\tau \sum_{(i \in N, j \neq CC)} c_{ij} S_{i,j} + \tau \sum_{i \in N, j = CC} (c_{ij} \sum_{k \in N} \frac{b_k X_{i,j,k}}{B_{i,j}}) + \sum_{i=1}^{N} IC \cdot Y_i \tag{4.11}$$

Eq. 4.11 computes the total cost as the sum of fixed operational costs for wireless and SCADA links, sum of link utilization dependent operational costs for cellular links and the sum of installation costs. Notice that a fixed operational cost is assumed for wireless and SCADA links, because irrespective of the link selected for sensor data transmission, these links are operational for other requirements such as data transmission among the sensors in the SAT. Nonetheless, the formulation can easily be modified if the operational costs for all the links need to be made utilization dependent. We analyse the impact of link utilization based costs in section IV F.

4.5 Link reliability

In the proposed formulation presented before, we assume perfect wireless link conditions. However, in reality, wireless links can be unreliable due to wireless interference, channel loss, multi path fading etc. [36]. In the following, we present a method to extend our existing formulation to address link unreliability.

Let Rel denote path reliability that can be specified as part of input requirements. This means that each flow must reach the control center, CC with probability Rel. The path reliability Rel is related to the constituent link reliabilities. Consider a link l with link reliability, ρ_l. For a path with n links, the reliability of the path can be calculated as ρ_l^n. Thus, for every flow the condition $\rho_l^n \geq Rel$ must be satisfied. Given link reliability and path reliability, n can be found offline as the highest integer satisfying $\rho_l^n \geq Rel$. This n represents the maximum number of links that a flow can traverse, in order to satisfy path reliability constraint. Thus, wireless link reliabilities can be easily addressed by our existing formulation with the addition of the following constraint,

$$\sum_{(i,j) \in E} X_{i,j,k} \leq n, \quad \forall k \in N \tag{4.12}$$

Eq. 4.12 ensures that the total number of links traversed by each flow, k must be less than or equal to n.

Link specific reliabilities can also addressed in our proposed formulation. However, it introduces a quadratic constraint modifying the linear optimization program to a quadratically constrained program. Specifically, consider that the link reliability of link (i, j) is denoted as ρ_{ij}. Then the Eq. 4.12 shown above will change to

$$\prod_{(i,j)\in E} \rho_{ij} X_{i,j,k} \geq Rel, \quad \forall k \in N \tag{4.13}$$

As evident, addressing link specific reliabilities results in a quadratically constrained optimization problem which is much harder to solve. Thus, link specific reliability can be addressed at the cost of higher complexity.

4.5.1 Constrained Cellular Coverage

In the previous section, we assumed that the transmission line considered is uniformly covered by a cellular communication network. This means that cellular transceivers can be placed on any tower. However, owing to the diverse geographical terrains traversed by the long transmission lines, there might be remote areas where cellular coverage is not available. Or there could be prolonged outage on certain cellular towers. In such cases, there are additional constraints on the placement of cellular transceivers. We call this version of the problem as *Coverage Constrained* placement problem where relay nodes can only be installed on a subset of the transmission towers which are covered by cellular service.

The *Coverage Constrained* placement problem formulation requires little modification from our original placement problem formulation. Let $V^{'}$ denote the set of nodes not covered by the cellular service. The constrained edge set, $E^{'}$ will be $E - \{(k, CC), \forall\ k \in V^{'}\}$. In order to model the *Coverage Constrained* placement problem, the input graph G needs to be replaced by $G^{'}(V, E^{'})$. Fig. 4.4 shows the example where towers 3 and 4 do not have any cellular coverage. The edges $(3, CC)$ and $(4, CC)$ are thus removed. In the formulation, the corresponding binary variables, Y_3 and Y_4 are removed to represent cellular unavailability at these towers.

Thus, with a simple modification in the formulation, restricted cellular availability can be easily addressed.

If there is a large segment of towers devoid of cellular coverage, then the responsibility of data delivery falls upon wireless Zigbee links. In such cases, the data delivery might not be able to meet the latency constraints given the limited capacity of wireless Zigbee links. Such cases can be easily identified though our proposed method as shown in Fig. 4.10.

4.5.2 Asymmetric Data Generation

The method proposed in [3] relies heavily on symmetry and hence cannot accommodate asymmetric flow bandwidth requirement in the network. There can be several scenarios leading to towers generating sensor data at different rates. This can be due to a requirement of fine grained sensor measurement in order to attain better situational awareness of a particular tower located in a sensitive area. Fig. 4.5 shows such a scenario. Our proposed formulation can easily accommodate such asymmetric requirements. This is because each flow b_k generated at a tower k is individually formulated. Asymmetric data generation will change the values of b_k and l_{ijk} in Eqs. 4.2 and 4.7. Hence the formulation solves both symmetric and asymmetric cases equally the only difference being in the input file specifying the flow requirements.

4.5.3 Incremental Deployment

With changing data traffic requirements and geographical expansion of the transmission line, the possibility of incremental deployments is always present. If at each such future requirement, an entirely new input is given to the optimization formulation without taking into account the currently existing cellular enabled towers, the solution may be a costlier deployment. We term this method as *memoryless deployment* because it discards any memory of existing cellular enabled towers. This method can result in installation on an entirely new

set of towers, thus risking loss of any investment made in installing existing cellular transceivers in the first place.

We present a method to add new cellular links on top of existing network to satisfy newer

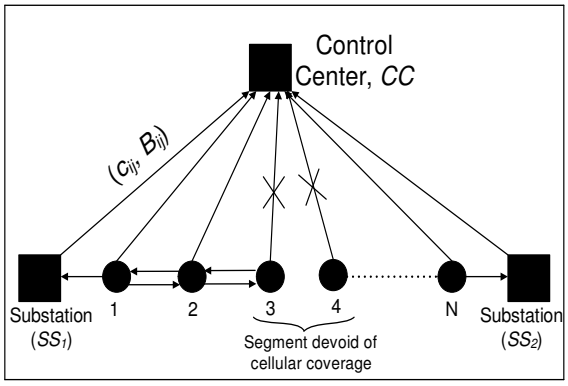

Figure 4.4: Placement graph for constrained cellular coverage

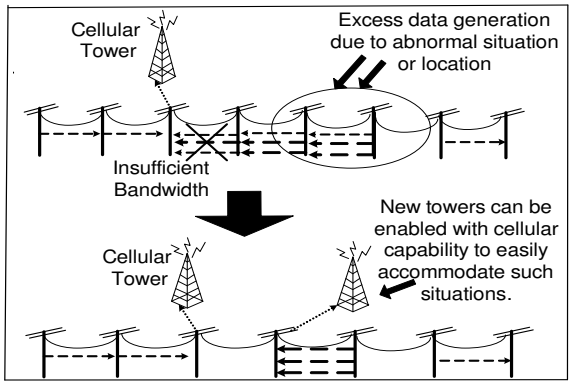

Figure 4.5: Accommodating asymmetric flow requirements

requirements while minimizing installation costs. In order to do so, only the input to the optimization program needs to be modified. The installation costs for currently deployed towers is made equal to 0 and flow requirements are increased as per new specifications leading to modified latency calculations. Since the installation costs of currently deployed towers is now 0, they will be automatically considered as cellular enabled towers in the optimized answer. The optimization formulation will make as much use of such existing cellular enabled towers as possible in minimizing total costs. Thus, *incremental deployment* reuses the existing cellular enabled transceivers as much as possible.

4.6 Performance Evaluation

We consider a transmission line network with 75 towers with an average span length of 800 ft [74]. In order to reflect real-world scenarios, the bandwidth of the optical fiber links (SS, CC) is taken as 10 Gbps. Bandwidth of the cellular links is taken as 75 Mbps and latency incurred in the cellular link due to state transition delay, access delay, and handover etc. is taken as 50 ms [75]. The bandwidth of the IEEE 802.15.4 wireless links is 250 kbps and latency incurred due to these links is 16 ms [76]. The length of the data packet generated by each tower is 32 kbits [19]. The performance metric of interest is the total cost of the network including the installation and operational costs. We consider three pricing schemes named C_1, C_2 and C_3 and analyse the effect of different costs associated with each type of link present

in our network. They vary in their ratio of operational costs attached to each type of link. A pricing scheme can be described as a ratio of operational costs of optical fiber to cellular to Zigbee. Thus, a scheme (1:10:2) would mean that operational cost of the three types of links: optical fiber, cellular and Zigbee are in the ratio 1:10:2.

We study several different scenarios including variation in flow bandwidth, end to end deadline and network size. We compare the results of our proposed formulation called Integer Linear Program (ILP) with the method proposed in [3] referred here as the Quadratic Equation method (QE). We also evaluate the cost of the network in cases of constrained cellular coverage and incremental deployment.

The proposed formulation provides an optimal solution to the wireless network design problem at the granularity of a transmission corridor as shown in Fig. 4.3. This solution can be independently computed for each corridor. While computationally expensive, these calculations are required to be done only during the offline centralized network planning stage. Thus, the fact that an ILP formulation is expensive to compute is mitigated by the small number of times such an optimization needs to be performed. However, for cases where the transmission corridor comprises of several hundreds of transmission towers, heuristic methods may be better suited. This is because even though they compromise on the accuracy of the result, they are capable of providing a solution with much less computing resource requirements. Although the ILP model is harder to scale and computationally-intensive, it is helpful in determining a lower bound on the costs. We use the ILOG CPLEX 12.2 software c̃iteCplex to solve the proposed Integer Linear Program (ILP). The complexity of the ILP can be estimated by the size of the input problem. For a 75 node network, there were 2600 binary variables and 2100 constraints. The simulations were run on Intel(R) Xeon(R) X5650@ 2.67GHz machines. The simulations took a minimum of 0.16 seconds and a maximum time of 4.25 hours.

4.6.1 Effect of variation in Flow Bandwidth

Fig. 4.6 shows the effect of the amount of data generated by each tower and its effect on the feasible operation of the transmission line. In this simulation, we consider a 75 node network

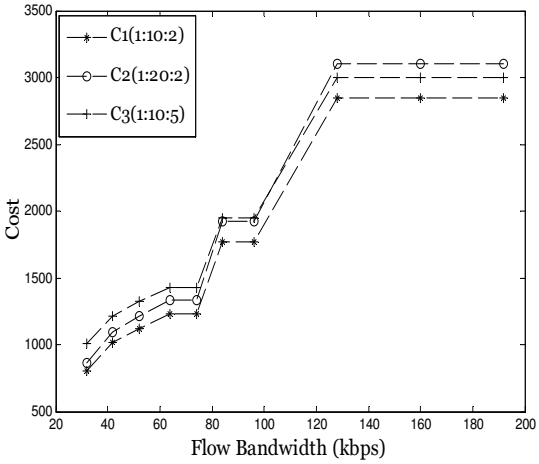

Figure 4.6: ILP: Effect of variation in flow bandwidth

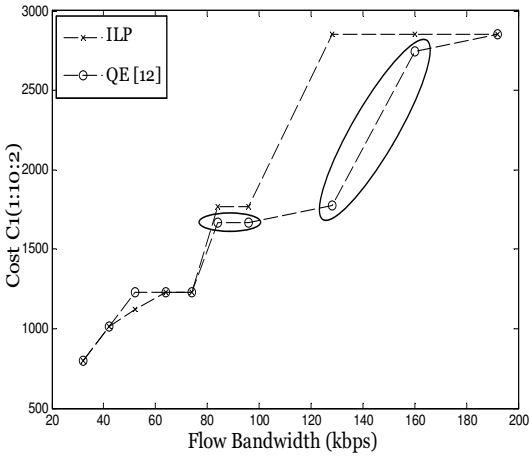

Figure 4.7: Comparing ILP and QE [3] with respect to varying flow bandwidth

with a deadline constraint of 3 sec. The packet size of the sensor data generated at each tower is the same. Given constant bandwidth of the wireless links, only a certain number of flows can be multiplexed on any link. Thus smaller the flow bandwidth requirement, more flows can be multiplexed on each link. Given a large deadline requirement, this results in reducing the number of cellular links to be used and hence the cost. The performance graph echoes this observation. Notice in the graph that for values of flow bandwidth greater than or equal to 128 kbps (specifically 128 kbps, 160 kbps and 192 kbps), the cost becomes constant. This is because given the wireless link bandwidth of 250 kbps, at most one flow can be multiplexed on each link. Thus the network design remains same for each of these three flow bandwidth requirements. Also in cases where, flow bandwidth is greater than wireless link bandwidth, then the remaining options are either deploying an all cellular or all wired solution.

Fig. 4.7 compares the results given by our proposed algorithm, ILP and the proposed method (QE) in [3] with respect to variation in flow bandwidth. As mentioned earlier, the QE method [3] utilizes a quadratic equation to obtain the number of cellular enabled towers. Roots of quadratic equation are rounded off to the nearest integer to depict the number of cellular enabled towers which must be an integer. This rounding off leads to incorrect results as can be seen in the plotted curves. Once flow bandwidth is greater than 128 kbps, the results should

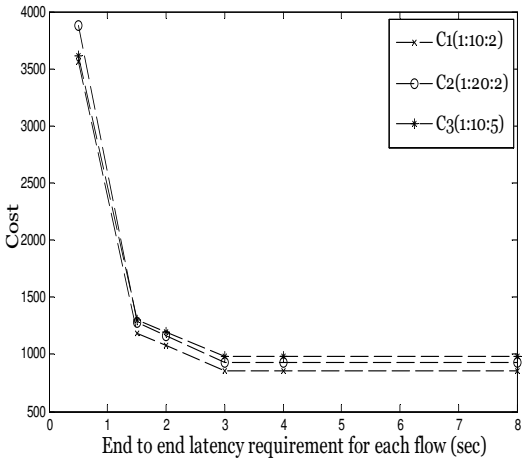

Figure 4.8: ILP: Effect of variation in end to end flow latency

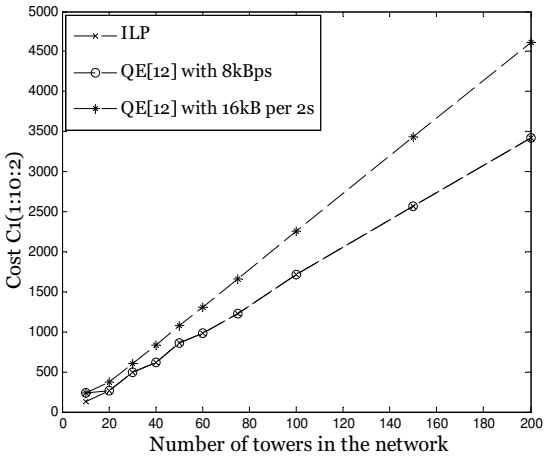

Figure 4.9: Comparing ILP and QE [3] with respect to varying number of towers

be the same for the three cases (128 kbps, 160 kbps and 192 kbps) as explained previously. However, the QE method tends to give incorrect results owing to errors encountered due to rounding of roots. Similarly for flow bandwidths of 84 kbps and 96 kbps, the cost incurred by QE method is less, but that is because number of towers selected by the QE method are insufficient leading to constraint violation.

4.6.2 Effect of variation in Flow Latency

Fig. 4.8 shows the effect of variation in end to end flow latency with respect to cost. We consider the time scale of one SCADA cycle which is 4-8 sec [4]. The results show that in cases of very stringent deadline requirement ($\approx$ 0.1s), a cellular transceiver should be installed on each tower. Thus every tower uses the cellular link to avoid any deadline miss hence ensuing a huge cost. In the given 50 node network, for relatively relaxed deadline requirements($\approx$ 2-4 sec) the lowest cost is attained by fully utilizing the wireless network. At the time scale($\geq$ 3 sec) the cost becomes constant because now the system is more bandwidth limited than latency limited.

4.6.3 Effect of Network Size

Fig. 4.9 shows the effect of variation in the number of transmission towers with respect to the cost. Given the linear structure of the transmission line, the cost increases approximately linearly with respect to the number of towers in the network. We assume all towers generate

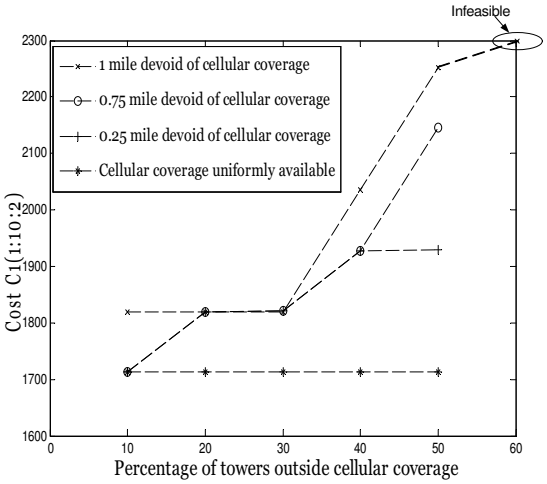

Figure 4.10: ILP: Capable of addressing variations in cellular coverage

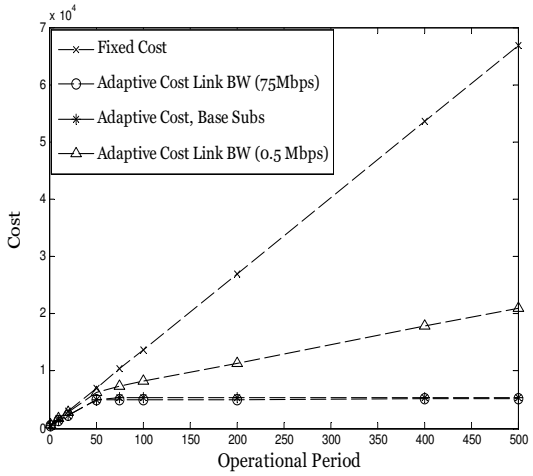

Figure 4.11: Effect of variation in operational period

64 kbits of data per monitoring cycle [19] and the end to end deadline is 8 sec. We compare the results of ILP with two data generation rates in QE one at 8kBytes per second and one at 16kBytes per 2 seconds both being equivalent to 64kbps. The QE solution results in higher cost for the case of 16kBytes per 2 seconds. For any given network, this graph can be used to find the most cost effective solution by plotting the various pricing curves.

4.6.4 Effect of variation in Cellular Coverage

Fig. 4.10 shows the effect of variation in cellular coverage and its effect on the installation and operational cost of the network. We consider a 100 node network with constant flow bandwidth of 64kbps and a deadline of 8 sec. We vary the percentage of nodes which can be devoid of any cellular coverage from 10% to 50% of the network size. Further, we vary the length of the segment of the transmission line devoid of any cellular coverage from 0.25 mile to 1 mile. This is done to mirror real world situation where it is more likely that adjacent

towers are devoid of cellular coverage rather than random single towers. For comparison, the results of cellular constrained scenarios are compared with the symmetric case where coverage is available throughout the network. In cases where a large number of towers are devoid of cellular coverage, there might not be any feasible solution as shown in the figure. An analysis of this nature helps in finding the feasibility of the wireless option in the cellular constrained areas.

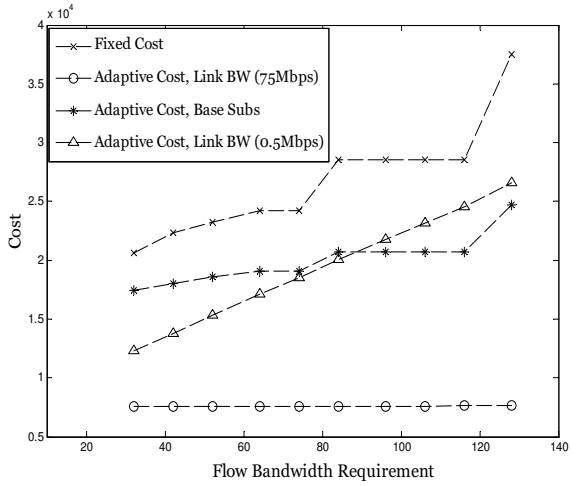

Figure 4.12: Effect of link utilization

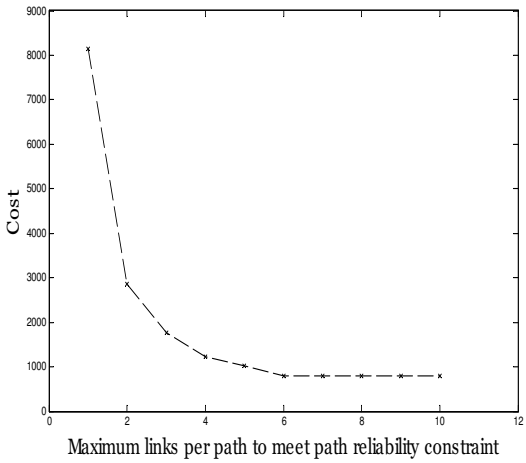

Figure 4.13: Effect of link unreliability

4.6.5 Effect of Operational Period

Fig. 4.11 shows the effect of operational period on the total costs (initial installation costs and operational costs over the operational period). In case of fixed operational costs, total costs increase rapidly with increasing operational period. In case of adaptive costs depending upon link utilization the total costs reduce dramatically. Also in this case, if the available link bandwidth is much higher as compared to bandwidth utilization, then operational costs turn out be very less and over time the initial investment on installation is recovered. Thus it can be profitable to deploy a cellular transceiver on a bigger subset of towers. However, if cellular bandwidth is relatively less, making the link utilization based operational costs a bigger part of the total costs, then the number of cellular towers needs to be optimized. This graph illustrates how different factors such as available link bandwidth, operational period and

cost ratios affect minimum cost network design.

4.6.6 Effect of Link Utilization based Cost

Fig. 4.12 shows the impact of link utilization based *adaptive* cost models on total network design cost. In this simulation, we consider a 75 node network with a operational period of 100. We consider 4 cost models explained as follows.

The *Fixed Cost* model considers fixed subscription cost ignoring adaptive link utilization. As evident, it incurs highest costs because the number of deployments increase with increasing flow bandwidth requirement.

The *Adaptive Cost, Link BW(75Mbps)* model is the link utilization based cost model where cellular bandwidth is 75Mbps incurring the lowest costs. The optimal solution here is to deploy all towers with cellular capability. This solution is optimal due to two factors. One, the subscription costs become negligible as compared to wireless operation costs given the high cellular bandwidth availability as compared to utilization. Secondly, a huge initial investment can be easily recovered given a large operational period. The negligible subscription cost acts as an incentive to initially deploy all towers with cellular capability.

In order to consider the effect of available cellular bandwidth, we consider an *Adaptive Cost, Link BW(0.5Mbps)* model with a smaller available link bandwidth of 0.5Mbps. In this case link utilization based subscription costs are non negligible resulting in increased cost.

We consider another cost model, *Adaptive Cost, Base Subs* where in addition to the link utilization based adaptive costs, a *fixed base subscription fee* is charged per active cellular link per period. The available cellular bandwidth is 75Mbps. Thus subscription costs are negligible but base subscription fee acts as a major factor in cost calculations. When compared with *Adaptive Cost, Link BW(0.5Mbps)*, the tradeoff can be observed. At lower flow bandwidth requirements, *Adaptive Cost, Base Subs* incurs more cost due to overpowering base subscription fee. But at higher requirement, link utilization based costs result in more costs by *Adaptive Cost, Link BW(0.5Mbps)*.

4.6.7 Effect of Link Unreliability

Fig. 4.13 shows the impact of link unreliability on network design cost. In this simulation, we consider a constant flow bandwidth of 32kbps and a deadline constraint of 3 sec. At lower link reliabilities, a path should consist of lesser number of links to maintain path reliability constraint. This leads to more cellular towers being deployed resulting in higher costs. As link reliability increases, more wireless links can be utilized resulting in cost reduction. After a certain point (maximum links ≥ 6), any further increase in link reliability does not affect cost reduction. This is because other optimization constraints such as limited bandwidth and latency limit the number of flows per link and hence the number of links per path.

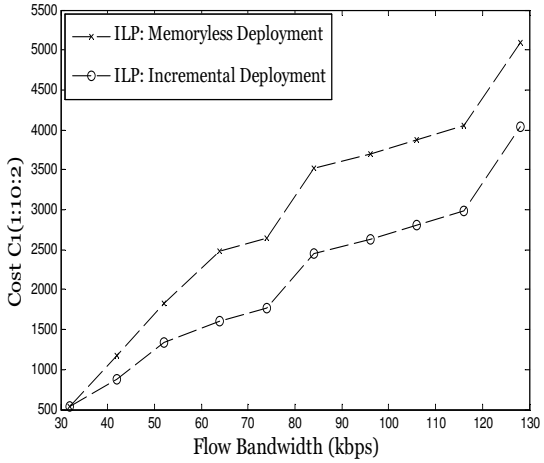

Figure 4.14: ILP: Capable of cost optimized incremental deployment

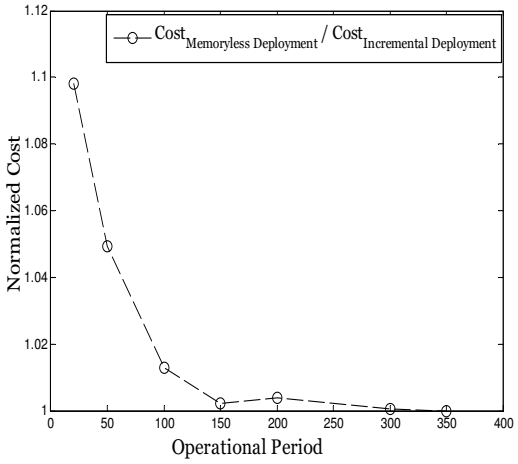

Figure 4.15: ILP: Comparing Memoryless deployment with Incremental deployment

4.6.8 Effect of Incremental Deployment

We perform experiments on a network of 50 towers with each tower generating sensor data at the rate of 32 kbps initially. The data generation rates are then gradually increased upto 128 kbps to mirror the increasing bandwidth demands in the future. Fig. 4.14 shows the cumulative costs incurred by the two methods: incremental deployment and memoryless deployment when the operational period is equal to one. Memoryless deployment starts with a clean slate each time a new requirement comes in. Due to this, memoryless deployment ends up installing a

cellular transceiver on a much bigger subset of towers. Incremental deployment modifies the input to the optimization and adds new links on top of existing network such that the existing design is utilized as much as possible incurring lesser deployment costs.

Please note that both deployment methods are variations of our proposed formulation producing optimal solutions depending on the input being provided to them. Our formulation offers the flexibility to perform different types of analyses. Factors such as operational period and link utilization based costs affect the performance of the two methods. Consider Fig. 4.15, where operational period is varied from 20 to 350. We consider a cost model (C1(1:10:2)) with the variation that each active cellular link is charged a base subscription fee and link utilization based subscription cost instead of fixed subscription cost. Each point on the graph, represents the total costs of an instance where a set of data requirements (starting from 32kbps to 128kbps) was operational for a period τ. Each value on the curve is the total cost of memoryless deployment normalized with respect to the respective value for incremental deployment to show their relatively close but different performance.

In Fig. 4.15, we observed that at low operational periods, memoryless deployment incurs more cost than incremental deployment. This is because memoryless deployment ignores any investment made in the earlier deployment and small operational period do not allow enough time to recover from that investment. Incremental deployment on the other hand, avoids extra installations by making the most use of the already deployed towers. At higher operational periods, the installation costs can be recovered and hence memoryless and incremental deployments produce solutions with equal costs.

4.6.9 Modeling multiple constraints

In the proposed formulation, we presented a generalised formulation in order to encompass varied application scenarios. However, it can be difficult to model every operational constraint. Mainly, the most important constraints that need to be modeled for a feasible network design are flow latency and flow bandwidth constraints. This is because Zigbee links can provide the least expensive communication, but their limited bandwidth proves to be a major bottleneck.

Also the topology of transmission lines presents critical challenges towards low latency communication being achieved solely by Zigbee links. These are followed by the cellular coverage constraint since extreme unavailability of cellular coverage might render the network design infeasible. Other constraints such as link reliability, link utilization based cost structure, asymmetric data generation etc. are less critical for a feasible cost optimized network design. The results we present in the chapter individually analyse the effect of variation in each of the constraint on the cost of the network while other constraint are assumed to take up average values. Specifically, we find that network design is feasible at very low costs at higher latency; at very high costs at higher flow bandwidth. Further, we find that if link utilization based cost structure is adopted then flow bandwidth requirement might have no effect on the total costs as shown in Fig. 4.12.

4.7 Conclusions

In this chapter, we presented a formulation for a cost optimized wireless network capable of transmission of time sensitive sensor data through the transmission line network in the presence of delay and bandwidth constraints. We formulated a placement problem to find the optimal locations of cellular enabled transmission towers. Our analysis shows that a transmission line monitoring framework using WSN is indeed feasible using available technologies. We compared behaviour of our proposed method with the method proposed in [3]. Our proposed formulation is generic and encompasses variation in several factors such as asymmetric data generation at towers, wireless link reliabilities, link utilization dependent costs, non-uniform cellular coverage characteristics and requirements for cost optimized incremental deployment. Our evaluation studies show that the main bottleneck in cost minimization is wireless link bandwidth. Further, in cases of increasing flow bandwidth, the limited wireless link bandwidth leads to a feasible but expensive design due to increased dependence on cellular network to satisfy constraints.

CHAPTER 5 Conclusions and Future Work

This dissertation addressed the problem of resource management in real-time Wireless Sensor Networks (WSN) for applications in Cyber Physical Systems. The goal was to design efficient offline and run-time energy-aware scheduling algorithms to effectively perform resource management in the presence of several constraints. Additionally, a network design solution for real-time monitoring of overhead electrical transmission power lines was proposed. Specifically, the following contributions were made:

(1) Offline energy-optimized schedule construction

1. The problem of joint scheduling of tasks and messages in a data collection tree network in the presence of precedence, deadline and interference constraints was mathematically formulated.
2. An efficient algorithm to generate a feasible, energy-aware schedule leveraging the energy delay tradeoff was developed [31].
3. The performance of the proposed algorithm was found comparable to that of the optimal solution obtained using a Mixed Integer Linear Program.

(2) Online scheduling for energy-aware adaptive Medium Access Control (MAC) protocols

1. Online adaptive algorithms were developed to exploit runtime variations for two categories of MAC schemes: Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).
2. For TDMA systems, lightweight node level slack management scheme was developed to exploit slack generated due to temporal correlation at runtime [77].

3. Simulation results show that significant reduction in energy consumption was achieved with the proposed schemes.
4. For CSMA/CA systems, a heuristic that adapts to varying load and channel state to obtain combined reduction in energy consumption and transmission delay was developed [78].
5. Simulation results show that our proposed load aware and channel aware algorithm significantly reduces energy wastage due to failed transmissions and results in higher transmission success rate.

(3) Wireless Network Design for Transmission Line Monitoring in Smart Grid

1. Application specific requirements were characterized and a hybrid network design problem was formulated for real-time monitoring considering the latency and bandwidth constraints of WSNs.
2. A placement problem was formulated using an Integer Linear Program to find the optimal location of cellular enabled transmission towers.
3. A generic formulation was devised that addresses real world scenarios with asymmetric sensor data generation, unreliable wireless link behavior, non-uniform cellular coverage etc [79].
4. Proposed formulation provides a method to address requirements for incremental deployment in the future in a cost minimized manner.
5. The proposed design of a high performance data communication network can prove immensely beneficial to electrical utility companies planning to either manage an existing transmission infrastructure, or deploying a robust new one.

Future Work

Our research can be expanded to address several exciting problems in the area of resource management in WSN and their integration in future applications. We propose some ideas for short term and long term future work as follows:

(1) Short term research directions

1. *Refined cost models for network design and planning:* The cost optimized network design proposed for transmission line monitoring can be extended to address the changing value of money over time. Specifically, we recognize that the proposed network design needs to remain relevant and cost optimized for long operational periods typically fifteen to twenty years in the future. A more elaborate cost model can be created by considering factors such as monetary inflation that result in increased price levels of goods (sensors, cellular transceivers) and services(sensor maintenance, cellular subscription) etc.
2. *Distributed algorithms:* Joint scheduling of tasks and messages can be performed through distributed algorithms to further reduce the energy overhead stemming from centralized operation. Designing distributed algorithms for a WSN with heterogeneous nodes and channel conditions presents several research challenges such as lack of global knowledge about the workload and channel condition, etc.
3. *Runtime Variations:* The effect of combining other runtime variations like dynamic execution time of computation tasks, channel variations, and exploiting both spatial and temporal correlation can be studied. The study of associated tradeoffs to characterize the net benefit and designing low complexity algorithms to leverage these factors forms an interesting research problem.

(2) Long term research directions

1. *Network Coding:* Given the expansive scale at which the WSNs of the future will be deployed and the associated increase in bandwidth requirements, the existing MAC and routing technologies will fall short in transporting the enormous amount of data in a reliable manner. Network coding brings the promise of providing significant throughput improvements in wireless networks by reducing the number of transmissions which has twofold benefits of energy reductions and throughput maximization. Network coding aided scheduling can prove to be highly energy efficient for both

broadcast and multicast scenarios. The investigation of tradeoffs between network coding based throughput maximization and energy minimization form an exciting research problem.

2. *Fault diagnosis:* Applications utilizing sensor networks to procure monitoring data must deal with unpredictability and unreliability in the data due to uncertainties in wireless channel and in sensor measurements. This calls for diagnosis of the fault scenarios to classify them as faults in the physical structure or in the monitoring framework for appropriate control actions to take place. A fault diagnosis framework addressing such uncertainties in sensor measurements to provide reliable structure monitoring forms an interesting future research area.

Bibliography

[1] H. Liu, A. Chandra, and J. Srivastava. esense: energy efficient stochastic sensing framework for wireless sensor platforms. In *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pages 235 –242, 2006.

[2] C. Schurgers, V. Raghunathan, and M. B. Srivastava. Power management for energy-aware communication systems. *ACM Trans. Embed. Comput. Syst.*, 2(3):431–447, August 2003.

[3] Y. C. Wu, L. F. Cheung, K. S. Lui, and P.W.T. Pong. Efficient communication of sensors monitoring overhead transmission lines. *Smart Grid, IEEE Transactions on*, 3(3):1130 –1136, September 2012.

[4] R. A. Leon, V. Vittal, and G. Manimaran. Application of sensor network for secure electric energy infrastructure. *Power Delivery, IEEE Transactions on*, 2007.

[5] Y. Yi, F. Lambert, and D. Divan. A survey on technologies for implementing sensor networks for power delivery systems. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –8, June 2007.

[6] W.C. Feng, E. Kaiser, W.C. Feng, and M.L. Baillif. Panoptes: scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(2):151–167, May 2005.

[7] CIAO, Critical Foundations: Protecting Americas Infrastructures,. Technical report, 10 1997.

[8] A. Zaballos, A. Vallejo, and J.M. Selga. Heterogeneous communication architecture for the smart grid. *Network, IEEE*, 25(5):30 –37, September-October 2011.

[9] US Department of Energy, Smart Grid System Report, July 2009.

[10] K. Moslehi and R. Kumar. A reliability perspective of the smart grid. *Smart Grid, IEEE Transactions on*, 1(1):57 –64, June 2010.

[11] S. Mukhopadhyay, C. Schurgers, and S. Dey. Joint computation and communication scheduling to enable rich mobile applications. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pages 2117 –2122, November 2007.

[12] Y. Tian and E. Ekici. Cross-layer collaborative in-network processing in multihop wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 6(3):297 –310, March 2007.

[13] Y. Tian, J. Boangoat, E. Ekici, and F. Ozguner. Real-time task mapping and scheduling for collaborative in-network processing in dvs-enabled wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 10 pp., April 2006.

[14] B. Gedik, L. Ling, and P.S. Yu. ASAP: An adaptive sampling approach to data collection in sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(12):1766 –1783, December 2007.

[15] C. Alippi, G. Anastasi, M. D. Francesco, and M. Roveri. An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. *Instrumentation and Measurement, IEEE Transactions on*, 59(2):335 –344, February 2010.

[16] I. Demirkol and C. Ersoy. Energy and delay optimized contention for wireless sensor networks. *Comput. Netw.*, 53(12):2106–2119, August 2009.

[17] R. Oliveira and I. Koutsopoulos. Queue and channel state awareness for maximum throughput access control in csma/ca-based wireless lans. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1 –6, April 2009.

[18] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, MobiSys '08, pages 118–131, New York, NY, USA, 2008. ACM.

[19] K.S. Hung, W.K. Lee, V.O.K. Li, K.S. Lui, P.W.T. Pong, K.K.Y. Wong, G.H. Yang, and J. Zhong. On wireless sensors communication for overhead transmission line monitoring in power delivery systems. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 309 –314, October 2010.

[20] J. Chen, S. Kher, and A.K. Somani. Energy efficient model for data gathering in structured multiclustered wireless sensor network. In *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, pages 8 pp. –388, April 2006.

[21] Y. Yi, F. Lambert, and D. Divan. A survey on technologies for implementing sensor networks for power delivery systems. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –8, June 2007.

[22] Y. Yi, D. Divan, R.G. Harley, and T.G. Habetler. Design and implementation of power line sensornet for overhead transmission lines. In *Power Energy Society General Meeting, 2009. PES '09. IEEE*, pages 1 –8, July 2009.

[23] G.S.A. Kumar, G. Manimaran, and Z. Wang. End-to-end energy management in networked real-time embedded systems. *Parallel and Distributed Systems, IEEE Transactions on*, 19(11):1498 –1510, November 2008.

[24] C. T. Yeh, Z. Fan, and R.X. Gao. Energy-aware data acquisition in wireless sensor networks. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1 –6, May 2007.

[25] C. Schurgers, O. Aberthorne, and M.B. Srivastava. Modulation scaling for energy aware communication systems. In *Low Power Electronics and Design, International Symposium on, 2001.*, pages 96 –99, 2001.

[26] S. Cui, A.J. Goldsmith, and A. Bahai. Energy-constrained modulation optimization. *Wireless Communications, IEEE Transactions on*, 4(5):2349 – 2360, September 2005.

[27] S. C. Ergen and P. Varaiya. Tdma scheduling algorithms for wireless sensor networks. *Wirel. Netw.*, 16(4):985–997, May 2010.

[28] W. Wang, Y. Wang, X. Y. Li, W. Z. Song, and O. Frieder. Efficient interference-aware tdma link scheduling for static wireless networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, MobiCom '06, pages 262–273, New York, NY, USA, 2006. ACM.

[29] Y. Wu, X. Y. Li, Y. Liu, and W. Lou. Energy-efficient wake-up scheduling for data collection and aggregation. *Parallel and Distributed Systems, IEEE Transactions on*, 21(2):275 –287, February 2010.

[30] J. Ma, W. Lou, Y. Wu, X. Y. Li, and G. Chen. Energy efficient tdma sleep scheduling in wireless sensor networks. In *INFOCOM 2009, IEEE*, pages 630 –638, April 2009.

[31] B. Fateh and G. Manimaran. Energy-aware joint scheduling of tasks and messages in wireless sensor networks. In *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pages 1 –4, April 2010.

[32] Y. Yu and V. K. Prasanna. Energy-balanced task allocation for collaborative processing in wireless sensor networks. *Mob. Netw. Appl.*, 10(1-2):115–131, February 2005.

[33] B. Zhang, R. Simon, and H. Aydin. Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling. *Industrial Informatics, IEEE Transactions on*, 9(1):514 –526, February 2013.

[34] IBM ILOG CPLEX Mathematical Optimization Technology. `http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/cplex-optimizer/cplex-performance`. Accessed: 16/02/2010.

[35] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388 –404, March 2000.

[36] J. G. Proakis. *Digital Communications, 4th ed.,*. McGraw-Hill.

[37] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wirel. Netw.*, 11(4):471–487, July 2005.

[38] Intel Xscale pxa27x,. `www.intel.com/design/intelxscale`. Accessed: 16/02/2009.

[39] G. Anastasi, M. Conti, M. Di F., and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7(3):537–568, May 2009.

[40] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 15(1):121 –144, quarter 2013.

[41] L. Chong, K. Wu, and J. Pei. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. *Parallel and Distributed Systems, IEEE Transactions on*, 18(7):1010 –1023, July 2007.

[42] S. Chatterjea and P. Havinga. An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks. In *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, DCOSS '08, pages 60–78, Berlin, Heidelberg, 2008. Springer-Verlag.

[43] D. Chu, A. Deshpande, J.M. Hellerstein, and Wei H. Approximate data collection in sensor networks using probabilistic models. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, page 48, April 2006.

[44] C. Wang, H. Ma, Y. He, and Xiong S. Adaptive approximate data collection for wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 23(6):1004 –1016, June 2012.

[45] L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575 – 578, 2002.

[46] Y. Yang, V.K. Prassana, and B. Krishnamachari. Energy minimization for real-time data gathering in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, 5(11):3087 –3096, November 2006.

[47] H. Jiang, S. Jin, and C. Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064 –1071, June 2011.

[48] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, pages 100–109, New York, NY, USA, 2007. ACM.

[49] Y. Shi, Y. Hou, J. Liu, and S. Kompella. Bridging the gap between protocol and physical models for wireless networks. *Mobile Computing, IEEE Transactions on*, PP(99):1, 2012.

[50] M.R. Garey and D.S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness,*. W. H. Freeman & Co., New York, NY, USA.

[51] Y. P. Huang, C. C. Hsu, and S. H. Wang. Pattern recognition in time series database: A case study on financial database. *Expert Syst. Appl.*, 33(1):199–205, July 2007.

[52] Curt Schurgers. *Energy-Aware Communication Systems*. PhD thesis, University of California, Los Angeles, Los Angeles, USA, 2002.

[53] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 95–107, New York, NY, USA, 2004. ACM.

[54] A. Jindal and K. Psounis. Modeling spatially correlated data in sensor networks. *ACM Trans. Sen. Netw.*, 2(4):466–499, November 2006.

[55] J.L. Hill and D.E. Culler. Mica: a wireless platform for deeply embedded networks. *Micro, IEEE*, 22(6):12–24, 2002.

[56] Labdata. `http://db.lcs.mit.edu/labdata/labdata.html`. Accessed: 30/09/2011.

[57] C. Zhihui and A.A. Khokhar. A channel reservation procedure for fading channels in wireless local area networks. *Wireless Communications, IEEE Transactions on*, 4(2):689 – 699, March 2005.

[58] Z. Xiaobo, W. Heping, and A. Khokhar. An energy-efficient mac-layer transmission algorithm considering fading channels for cluster-based sensor networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1 –5, 30 2008-December 4 2008.

[59] S. Lanzisera, A.M. Mehta, and K.S.J. Pister. Reducing average power in wireless sensor networks through data rate adaptation. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1 –6, June 2009.

[60] IEEE 802.15, part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs), ANSI/IEEE, Standard 802.15.4 R2006, 2006.

[61] IEEE 802.15 Task Group 4b,. `TG4bcontributions",http://grouper.ieee.org/groups/802/15/pub/TG4b.html`.

[62] Z. Pei, L. Fangxing, and N. Bhatt. Next-generation monitoring, analysis, and control for the future smart control center. *Smart Grid, IEEE Transactions on*, 1(2):186 –192, September 2010.

[63] V. C. Gungor and F. C. Lambert. A survey on communication networks for electric system automation. *Comput. Netw.*, 50(7):877–897, May 2006.

[64] P. Ramachandran, V. Vittal, and G.T. Heydt. Mechanical state estimation for overhead transmission lines with level spans. *Power Systems, IEEE Transactions on*, 23(3):908 –915, August 2008.

[65] S. Malhara and V. Vittal. Mechanical state estimation of overhead transmission lines using tilt sensors. *Power Systems, IEEE Transactions on*, 25(3):1282 –1290, August 2010.

[66] J. Ausen, B.F. Fitzgerald, E.A. Gust, D.C. Lawry, J.P. Lazar, and R.L. Oye. Dynamic thermal rating system relieves transmission constraint. In *Transmission Distribution Construction, Operation and Live-Line Maintenance, 2006. ESMO 2006. IEEE 11th International Conference on*, October 2006.

[67] V.C. Gungor, L. Bin, and G.P. Hancke. Opportunities and challenges of wireless sensor networks in smart grid. *Industrial Electronics, IEEE Transactions on*, 57(10):3557 –3564, October 2010.

[68] S. Ullo, A. Vaccaro, and G. Velotto. The role of pervasive and cooperative sensor networks in smart grids communication. In *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, pages 443 –447, April 2010.

[69] A. Bose. Smart transmission grid applications and their supporting infrastructure. *Smart Grid, IEEE Transactions on*, 1(1):11 –19, June 2010.

[70] Load Cell Technology in Practice, Revere Transducers, March 2009. `http://www.societyofrobots.com/robottheory/loadcellprimer.pdf`.

[71] R.G. Olsen and K.S. Edwards. A new method for real-time monitoring of high-voltage transmission-line conductor sag. *Power Delivery, IEEE Transactions on*, 17(4):1142 –1152, October 2002.

[72] Sagometer, An EPRI Sponsored Technology,. Technical report, 2010.

[73] S. Gumbo and H.N. Muyingi. Performance investigation of wireless sensor network for long distance overhead power lines; mica2 motes, a case study. In *Broadband Communications,*

Information Technology Biomedical Applications, 2008 Third International Conference on, pages 443 –450, November 2008.

[74] American Transmission Company Technical Report,. `http://www.atcllc.com/pdf/MorganWernerWestapplication.pdf`. Accessed: 21/01/2012.

[75] Overview of the 3GPP long term evolution physical layer,. `http://www.freescale.com/files/wirelesscomm/doc/whitepaper\/3GPPEVOLUTIONWP.pdf`. Accessed: 30/09/2010.

[76] IEEE Standard for information technology-Part 15.4: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (LRWPANS). *IEEE 802.15.4-2006*, 2006.

[77] B. Fateh and M. Govindarasu. Energy minimization by exploiting data redundancy in real-time wireless sensor networks. *Ad Hoc Networks (Preprint)*, 2013.

[78] B. Fateh and M. Govindarasu. Energy-aware adaptive mac protocol for real-time sensor networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, 2011.

[79] B. Fateh, M. Govindarasu, and V. Ajjarapu. Wireless network design for transmission line monitoring in smart grid. *Smart Grid, IEEE Transactions on (Preprint)*, 2013.